任天堂

FAMILY COMPUTER™
ファミリー コンピュータ™

AND

FAMILY BASIC™
ファミリー ベーシック™

PERFECT BOOK

森田将史＝著

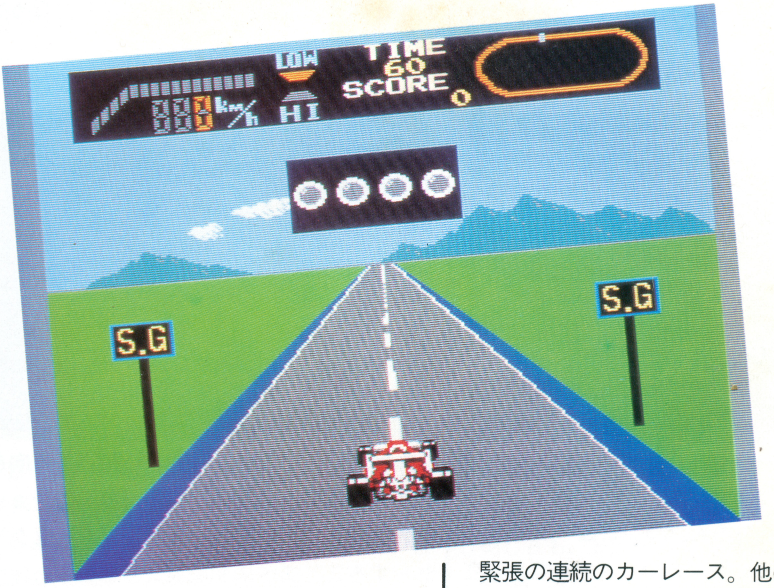


ファミリーコンピュータ™&ベーシック™パーフェクトBOOK

ナツメ社

GAME GALLERY

任天堂ファミリーコンピュータ・ビデオゲーム・カタログ(市販品のビデオゲームです)



F１レース

★★★★★★★★★★★★★★★★★★

うなりを上げてつっ走るF１マシン。君はこのレースにいのちを賭けるレーサーだ。

Aボタンはアクセル。押せばいよいよスタートだ。一瞬、タイヤが白煙を上げる。エンジンの回転数が上がってきたら、ギアをハイに切り換える。時速はすでに300Km/hを超えている。

やがて、前から先行車のグループが見えてきた。巧みにハンドルを切って追い越しをかけるが、敵も時々こちらに寄せてくる。Bボタンのブレーキをかけ、やりすごしたら、またアクセル。次はカーブだ！

緊張の連続のカーレース。他のレースカーや、コースの外側にあるクイにぶつかると大破するし、決められた時間内にゴールしないとゲームオーバーになる。

コースは、クリアするごとにむずかしくなる。最初に、初級・中級・上級のうちから選ぶこともできます。

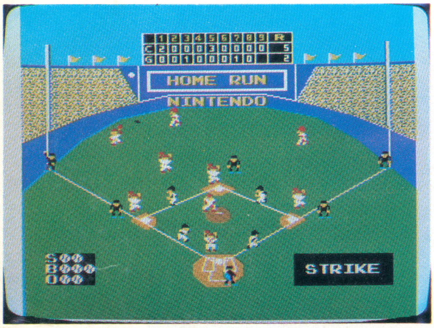
同じコースでも、プレイしていくと、昼、夕方から夜へと色が変わる。これは楽しい美しい。リズムの良い音楽がBGMだから、プレイするのも最高の気分！

走り出す際は、かならずギアをローに入れておくこと。これが、ポイントだ。そして、カーブの手前ではかならず減速すること。曲がりきれずに大破してしまいます。

とにかく、最高にノレるゲームです。おすすめのゲームカセット。

ベースボール

全天候型、オールシーズン・プレーの野球ゲーム。９回裏逆転サヨナラホームランの感激



いっけん地味なゲーム。実はこれぐらい大勢で集まってできて、なおかつ興奮するゲームも珍しい。もちろん、コンピュータ相手に１人でも、画面をはさんで２人でもできる立体野球ゲーム。

ゲームのスタートはまずチーム選びだ。どんな特徴のチームにするかを、６種類のうちから選んだら、先攻後攻が自動的に決まるから、プレイ開始。

攻撃の時は、バッターのポジションはもとより、ランナーの走塁指示から盗塁、バントもできます。守備にまわると、ピッチャーの球種（４種類）やけん制。さらに送球指示もできるのです。

残念ながら、延長戦はありませんが、とにかく、ハイテクニックで競う野球ゲームであることには間違いありません。ガンバッてください。



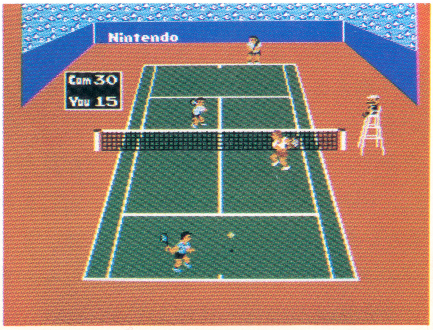
ゴルフ

たかがゲーム。だが、終いには本気になってしまう。毎日の努力が上達の秘訣だ。



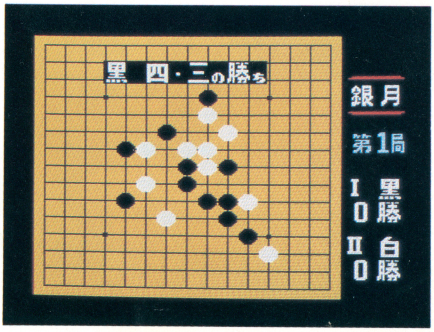
反射神経だけではゴルフはできない。風向きや地形の状態、クラブの種類、そしてショットの強弱と、これだけの要素をきちっとクリヤーすることでベスト・スコアも出るわけです。だからゴルフは読みのゲーム。頭を使わないと、ただみっともないスコアが表示され、アウトになってしまいます。

親父にだけ楽しませておくのはもったいない。いつかグリーンに出る日を夢見て、ひたすらに技をみがいておこう。



五目ならべ

説明する言葉もないくらいポピュラーな遊びだ。ただしコンピュータ相手だとこうなる。



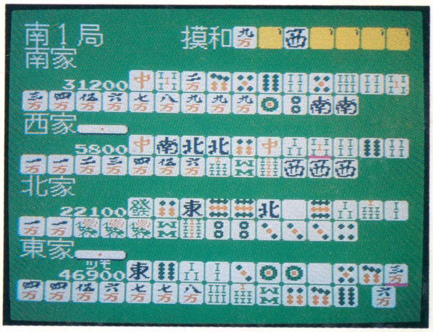
言ってしまえば、自分の色の石コロを５つタテ・ヨコ・ナナメ、いずれの方向かに並べてしまえばいいというゲーム。ただ、コンピュータが相手だと簡単にはいかない。

局面が進むにつれて、やたらむずかしくなって、最後はいつもパニックになる。

じっくり、腰をすえてやらなければ勝つことはできない。インテリジェンスの問われるゲームなのだから。いかにコンピュータの先を読むかが勝敗の鍵です。

４人打ち マージャン

メンツがなかなか４人集まらないのがマージャンだけど、これならいつでも楽しめる。



この麻雀はハンパじゃない。そのへんにころがっているマイコンの麻雀と比べても、速さも美しさも、比較になりませんよ。

絶対にお買い得です。ロンやチーができるパイが来ると自動的に表示して入力を待つほかに、食い断の指定やチョンボの判定まで、自動的にやってくれる。

また、あのパイをかき回すジャラジャラいう音もなくて、静かに勝負師の気分を味わうことができます。

テニス

強くなりたい。だれよりも。と、思いつつマシンに負けることあり。

本格派３Ｄテニスは、ボールから何から、すべて立体的に表示されるので、迫力も楽しさも段ちがい平行棒です。

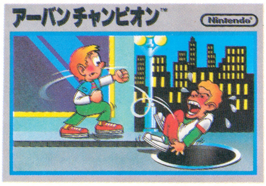
ポイントは、すべてボールにラケットを合わせるタイミング。これにつきます。はじめは、タイミングがうまく合わずに、空振りや身体にボールが当たってしまうなど、なかなかうまく行きませんが、ラリーが続くようになると、ヤメラレナくなります。もうこれはスポーツです。

テニスをやってなくても知らなくても、これは関係なく面白いのです。

アーバンチャンピオン

殴られるのは痛い。体も痛いが、なにせ心がいたくなる。なんの因果かこの珍商売。





きみはストリートファイターという言葉を知っているだろうか。それは、路上でケンカ試合をして見物人からお金をもらう人たちのことなのです。大変な商売もあったもんだね。

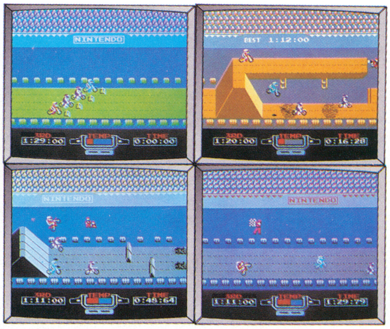
そして、きみはストリートファイターとなり、次々と現れる挑戦者をノックアウトしていくのである。その場合、逆のことも往々にして起こるのです。

時々落ちてくる植木鉢を寄けたり、じゃまなパトカーから逃げたりしながら、顔面やボディをガードしつつ、２種類のパンチを繰り出して、相手を落とせば勝ち。きみを紙吹雪が祝福してくれるだろう。

本当は、暗〜い路地や空き地でやるのだけれど、ゲームになるとやたら明るくなりました。出るか、必殺右ストレート！

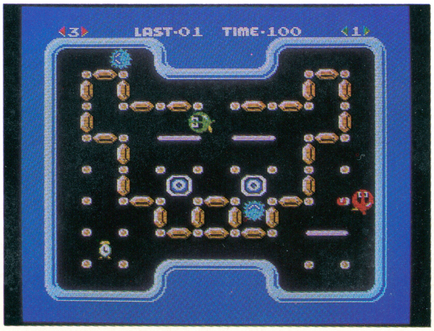
エキサイトバイク

ファミリーコンピュータ究極のゲームカセットがこれ。エキサイトしっぱなし。



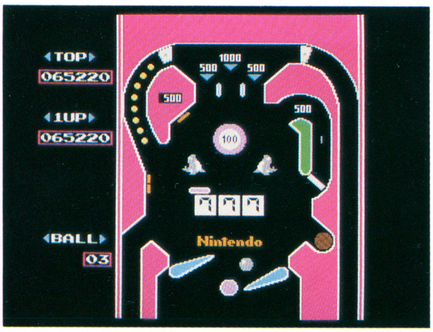
モトクロスです。ただ速く走ればいいってもんじゃない。ぬかるみや障害、悪路に加えて他のレーサーの妨害があって、まともになんか走れない。きみのテクニックで危機を乗り越えるしかない。

１人で走ることもできるので、レースに出る前に練習ができる。これはいい。またオリジナルコースが作れます。友達とタイムを競うのもいいですね。終場面の表彰台に、はたしてきみは上がることができるか！



ピンボール

ゲームセンターの片隅に、ついもひっそりとピンボールはある。そして今。



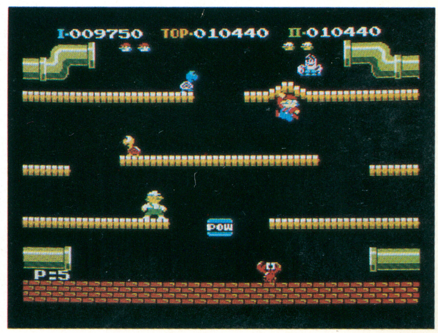
ただ単に、球をはじけばいいのかと思ったら大間違いなのだ。まず、台が２種類あり、片方は中央にスロットマシンが入っている。「７７７」を出すと高得点が出る。まるでパチンコだという人もいる。

もう一方はポーカーの要領で、カードを並べて高得点を目指す。

さらに、ボーナスステージには、うわさのマリオがレディを助けに行くという、１つで３倍楽しめるゲームです。

マリオ ブラザーズ

あの、うわさのマリオとルイージが帰ってきた。ただ、何がウワサか誰も知らない。



画面右上にあるパイプから、カメやカニ、はてはハエやフリーズと呼ばれる化物がゾロゾロ出てくる。攻撃をかけてくる。

下から敵をつっつくと気絶するので、それをキック！　こうして、全部の敵をやっつけると１面クリア。時々出てくるコインを取ると得点アップ。ボーナスステージでは、時間内にコインを取ると、取った数に応じてボーナスポイント。ファミリーコンピュータ・ゲームの傑作の一つ。

クルクル ランド

ゲームをしながら騒ぎたい。と、もし思う方がいたとしたら，これおすすめします。

とにかく、ピンのたくさん並んだ通路を走りまわって、通路に隠してある金塊を探し出す。きみの操るグルッピーは、ピンに手をかけるか、壁にバウンドするしか方向を変えられない。

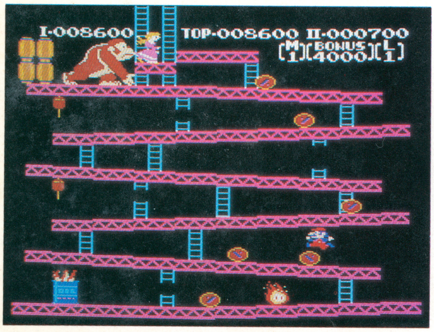
怪物のウニラやブラックホールが行手を妨げ、ウニラの仕掛けたラバートラップとかいうものが隠してあったりで、１面クリアするのは大変なことになってしまう。

できればこのゲーム、２人でプレイしたいものだ。やたら楽しい。隠しキャラも突然顔を出すしニギヤカである。ワイワイ遊ぶには最高のゲームなのです。

ドンキーコング

★★★★★★★★★★★★★★★★★★

ゲームセンターで有名な、うわさのドンキーコングが、ここにあります。

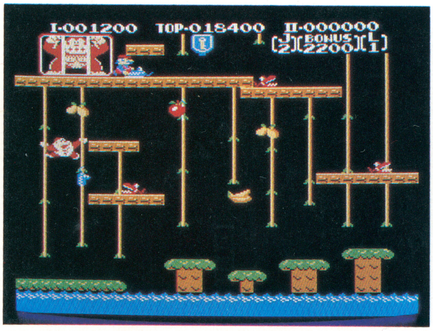


キングコングの孫のドンキーコングがレディをさらって、建築中のビルの頂上へ逃げてしまった。きみは、レディを助けるため、コングの落とすタルやおじゃま虫を、ハンマを使い、ジャンプでかわし、そのビルの頂上目指し登ることになる。場面は３つです。Ｉ・２面はビルの頂上へ登ればクリア。３面目はビル通路のボルトを抜いていく。全て抜くとビルはコングを道づれにつぶれていく。そういう美女と野獣の楽しいゲームなのです。

ドンキーコング Jｒ

★★★★★★★★★★★★★★★★★★

親父のかたきを打つために、けなげな息子がやって来て、また、やられる。ああ悲しい。

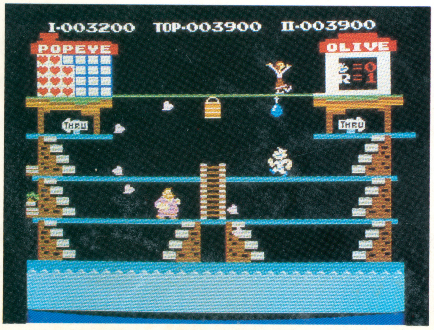


ワッハッハ、お笑いだ。前に悪さをしていたドンキーコングはマリオにつかまっていたのですね。

けなげにも息子のＪｒ君、親父を助けようと、必死の救出作戦を展開するのですが。

しかし、このマリオ君も意地が悪いね。赤ワニや青ワニを使って妨害するのよね。

さて、果たしてドンキーコングＪｒ君、守備よくお父さんを助けられますか！　前作ドンキーコングとひと味ちがったゲームです。



ポパイ

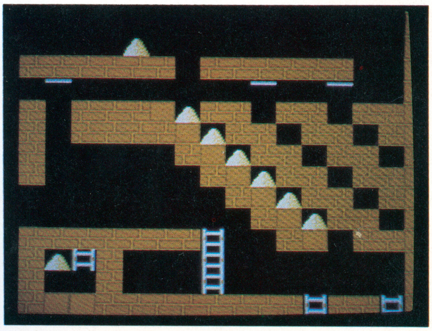
★★★★★★★★★★★★★★★★★★

腕にはイカリの入れ墨。懐にはホウレン草。つまりポパイといいたいのである。

なつかしい。ホウレン草のかんづめを片手に、ブルートを相手に、オリーブの取り合いをするポパイはなつかしい。

チャンピオンシップ
ロードランナー

世界のゲームマニアを魅了したロードランナーの上級生。興奮熱狂。唖然。騒然。



ハドソンから「ロードランナー」にひきつづき発表されたこのゲーム、まず「ロードランナー」を制覇してからチャレンジすることだ。とにかく上下左右にスクロールするステージが５０面。最初にこれをやったら、絶望のどん底に蹴落とされることを約束しよう。

「ロードランナー」「チャンピオンシップロードランナー」この２作を揃えてこそ、真のファミリーコンピュータ・マニアといってても過言ではない。絶品。ですね。

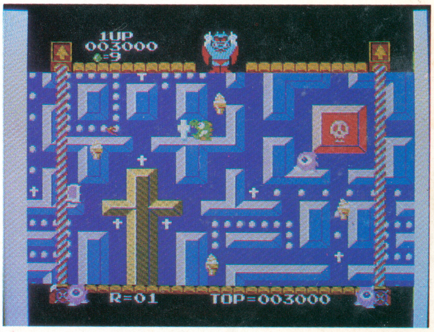
「私の愛の印よ」とばかりにオリーブが投げるハート。ブルートに取られてたまるかとポパイはせっせと集めて回ります。

ここでも、やっぱりブルートは悪人で、ジャマばかりする。これは変わることのない筋書きです。

しかし、ここにはホウレン草がない。ここがいつもと違う場面だ。そして、ホウレン草がない以上、ここは逃げの一手ですね。

デビル ワールド

どこでも地獄はつらいもんだ。たとえばゲームの中の世界にしても。



きみは怪獣タマゴンです。そして、運悪くデビルワールドの迷路地獄に迷い込んでしまったのです。意地悪なメダマンやボンボン、子デビルが追いかけて来ます。

この世界の支配者であるデビルまでが、「よいひまつぶしだ」とばかりに、迷路を囲む壁を動かし、メダマンをつぶそうとするのです。

そこできみの使命はといえば、「十字架を手に、迷路中に散らばるボワボワを集める」のです。十字架を持つ間は火が吹けますが、時間と共に十字架は消えてしまいます。



ボワボワを拾い終わったら、それからが勝負。デビルをやっつけるため、４スミにある聖書を集めて、これでようやく１面クリアとなる。理屈ぬきで楽しめる、かわいいゲームといっておこう。

ダックハント

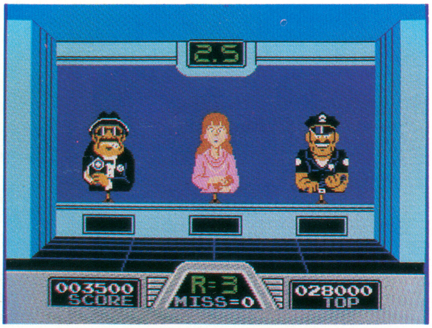
いくら撃ち落としても減らないダック。腕が悪いという原因だけがはっきりしている。



博愛主義者だから、「鳥なんか射ちたくない」という人は止めればいい。けれど銃はカッコがいい。男のロマンがある。そんな向きには最高のゲーム。犬が草むらに飛び込むと鳥が飛び出すので、すかさずＨＩＴ！　２人でプレイするときは、片方が鳥をコントロール。こうなると、お互い騙し合いですね。

ホーガンズアレイ

光線銃シリーズの新作ゲーム。ここは西部。開拓時代のお話である。



警察の射撃大会にあるそうだ。どこから誰が出てくるのかわからないが、その中から悪人だけを射つ。非情のガンマン。おっと、警官射っちまった。ちょっとキザかね。

２つのモードの他に、飛んでくる空カンを射って得点を競うトリックショットというゲームも楽しめます。



ワイルドガンマン

ここは無法の街。信じられるのは自分の力だけ。前にあるのはブラウン管。

賞金かせぎのきみは、高額の賞金のかかったならず者を訪ねて旅を続けている。これは光線銃シリーズ第１段。任天堂おてのもののゲームなのである。モードによっては１対１や２対１の対決が選べ、相手の賞金額によって腕の良さも変わる。ギャング相手に射ち合いもできます。

FAMILY COMPUTER FAMILY BASIC PERFECT BOOK

ファミリーコンピュータ ファミリーベーシック パーフェクトブック

はじめに

ファミリーコンピュータは家庭用のテレビに接続して使うビデオゲームマシンとしては他に類を見ないほど爆発的な売れ行きとなりました。

思うに、ファミリーコンピュータは、価格が安い、性能がいい、おもしろいゲームができるなどもろもろの理由から売れるべくして売れた、というところではないでしょうか。

ファミリーコンピュータが発売されたころに数多くのビデオゲームマシンがありましたが、今、店頭ではほとんどその姿を見ることはありません。ファミリーコンピュータがいかに売れ続けてきたのかを明確に物語っているといえます。

さて、ファミリーコンピュータは本体とゲームのカセットだけでも十分に楽しめますが、外部拡張コネクタに接続して遊ぶものも出ています。その第一段が光線銃、そして第二段として、これから紹介するファミリーベーシックが出されたわけです。

ファミリーベーシックはゲームと違い、「自分で造る」楽しみが味わえます。もしゲームで遊ぶことにあきてしまっているのなら、ファミリーベーシックを使って自分で何かを造る楽しみを味わってみたらいかがでしょうか。

そのとき、この本はきっと役に立つはずです。

もくじ

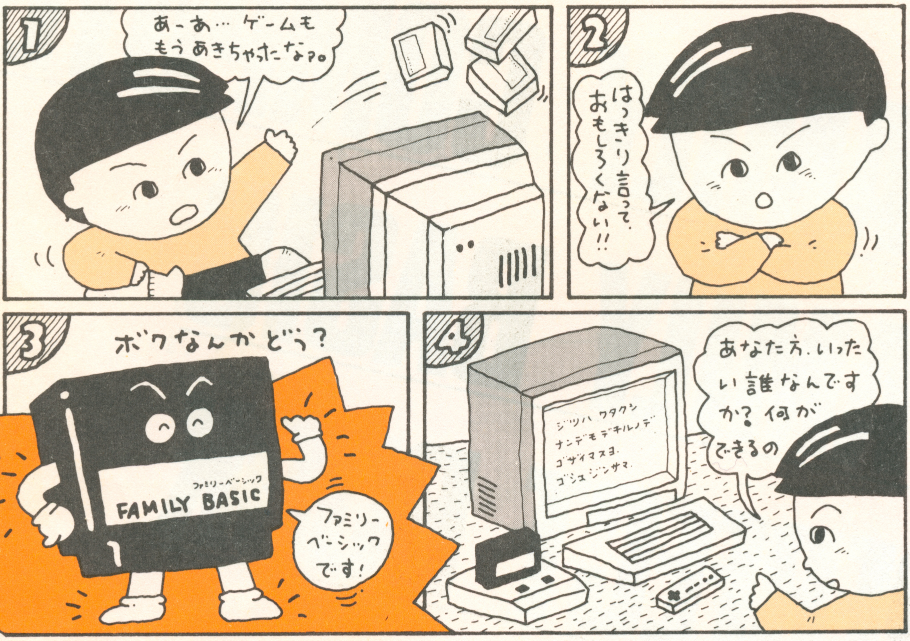

166ページには
ベーシックのコマンドや
ステートメントの読み方が
のっているよ!
167ページは
ファミリーベーシックV3.0の
紹介がのっているよ!

FAMILY COMPUTER FAMILY BASIC PERFECT BOOK

PART 1

ファミリーベーシックとは

ファミリーベーシック is
ザ・エンターテイメント・ベーシック



ファミリーベーシックはゲームよりおもしろい!

ファミリーコンピュータのゲームカセットはどれもオモシロイものばかりです。でも同じゲームで何回も遊んでいたらあきてしまい、またすぐに新しいゲームが欲しくなってしまうはずです。

ところがファミリーベーシックなら、いろいろなことに使えるので、あきるどころか使えば使うほど、ますますオモシロサが増してくるのです。もちろんゲームだって自分のアイディアで無限に作り出せてしまうのです。

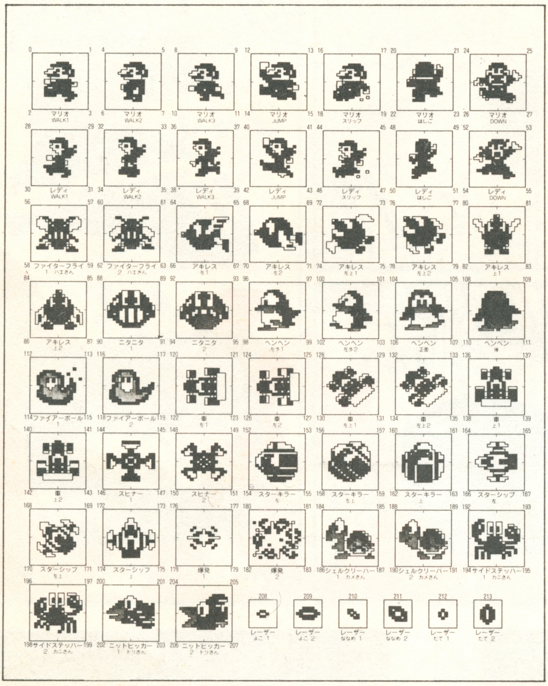
何・が・や・り・た・い

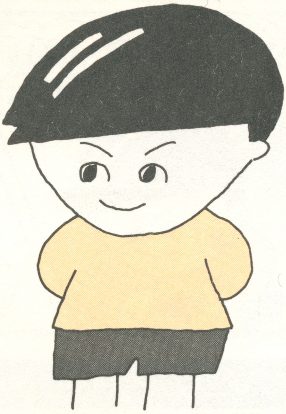
- ●ゲームを作りたい
- ●コンピュータに計算をやらせたい
- ●オリジナルの曲を作りたい
- ●メモ帳がわりにメッセージを記憶していてもらいたい
- ●明日のコンディションを占ってほしい



ファミリーベーシック・な・ら	こ・ん・な・こ・と・が・で・き・る
●ゲームのプログラムが組めるBASIC、ゲームの背景が作れるBG GRAPHICがあるよ	●オリジナルのゲームがたくさん作れました。
●計算ができるカリキュレータボードがあるよ	●電卓がいらなくなりました。
●簡単に音楽が作れるミュージックボードがあるよ	●イメージしたものをすぐに曲にできるようになりました。
●メッセージが記憶できるメッセージボードがあるよ	●家族とのコミュニケーションの機会が増えました。
●コンピュータが占いをしてくれるよ	●失敗することがなくなりました。

ゲーム作りラクラクのGAME BASIC





●アニメキャラクタが簡単に動かせるファミリーベーシックは……

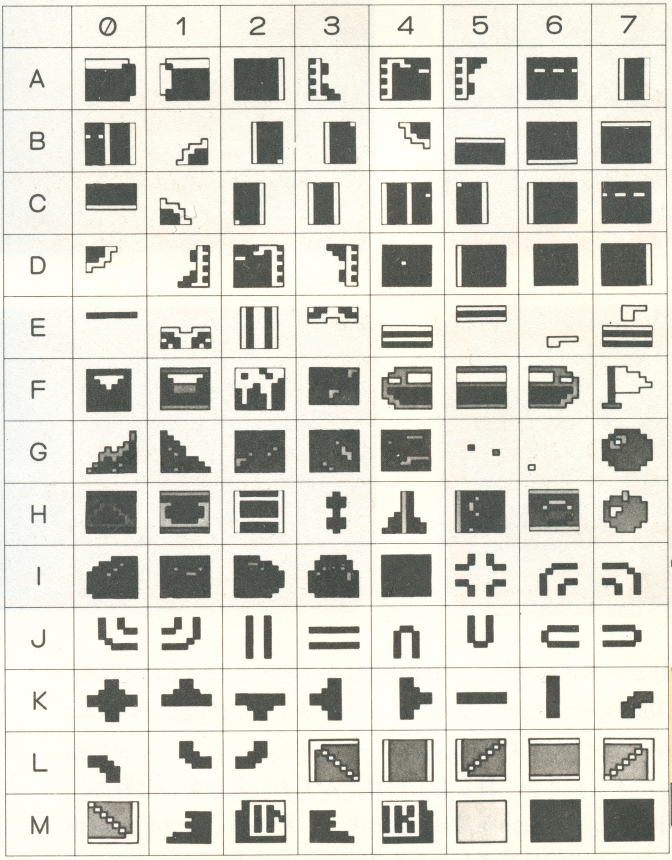
ファミリーベーシックに内蔵されているBASICは上にあるようなアニメキャラクタを簡単な手続きで画面に表示させることができます。

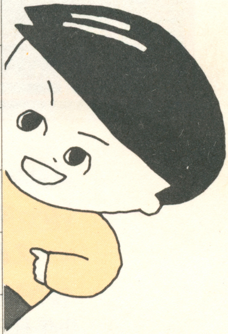
ですから、他のBASICのような「1ドット」単位で表すアニメキャラクタの形を考えるなんてことは全く必要ありません。

さらにこれらのアニメキャラクタを、動作をつけながら好きな方向へ動かすことがプログラムを作ったことのない人もあっという間にできてしまうのです。

くふうしだいでコントローラーにアニメキャラクタの動きを対応させるとか、アニメキャラクタの動きに合わせた音楽をつける、なんてこともできます。

●背景がキマレばゲームも盛り上がる



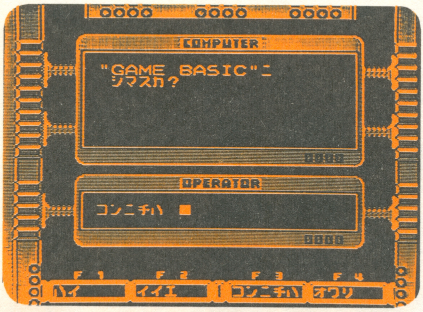


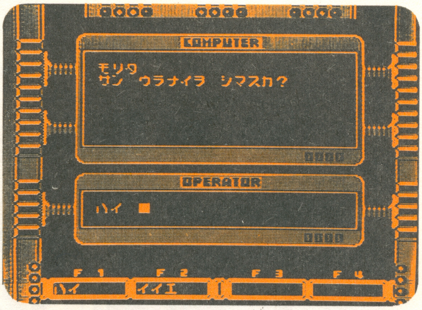
普通は難しい背景作りも BG GRAPHIC を使えばパターンの組み合わせだけでできます。パターンはあらかじめ104種類用意されていてそれぞれの好みの色を付けられます。

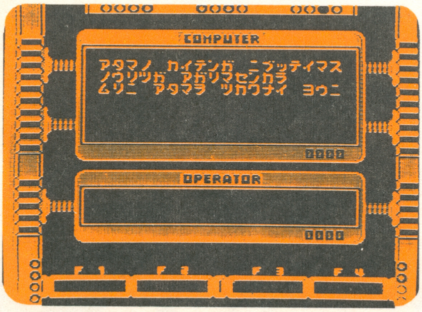
さらに、BG GRAPHIC で作った背景は BASIC から簡単に呼び出せますので、さきほどのアニメキャラクタと合成して画面に表示させればゲームなどの雰囲気を盛り上げることができるでしょう。

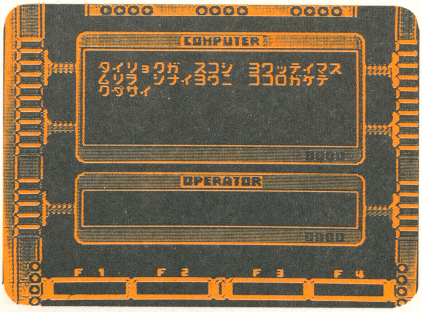
おもしろソフトいろいろのファミリーベーシック

コンピュータ占い





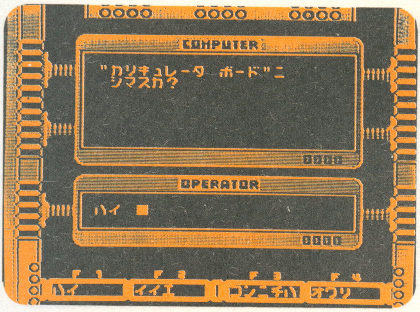


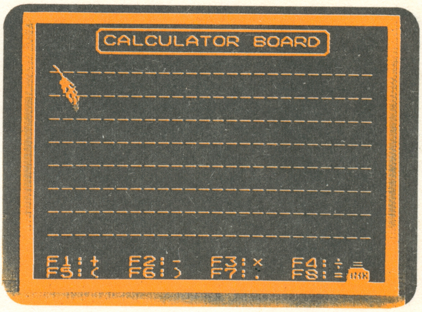


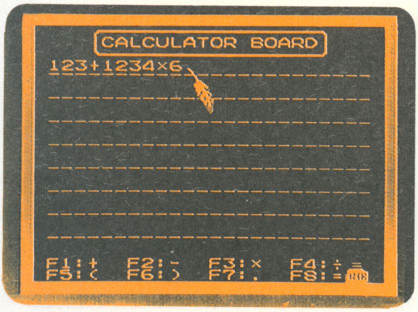
「コンピュータ占い」というのは今ではもうめずらしいものではなくなってしまいましたが、このファミリーベーシックにも「コンピュータ占い」が入っています。

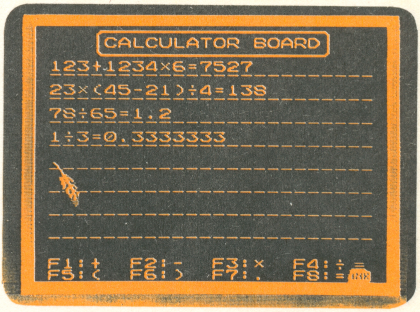
「コンニチワ」とか「HELLO」なんてあいさつをしたあと、生年月日と占ってほしい年月日を入れると、その日のコンディションと生まれてから何日経過したかを教えてくれます。

カリキュレータボード







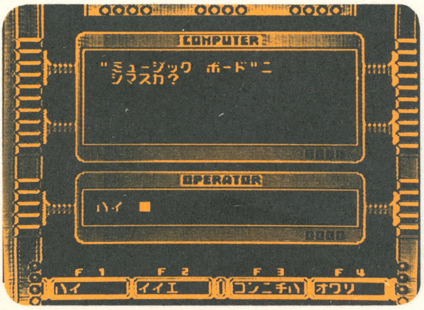


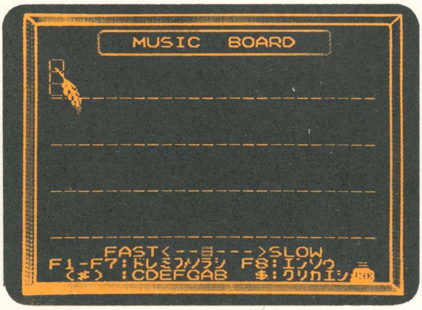
コンピュータというのは計算がいちばん得意です。ファミリーベーシックに内蔵されている「カリキュレータボード」は、入力した計算式をコンピュータが計算してくれます。

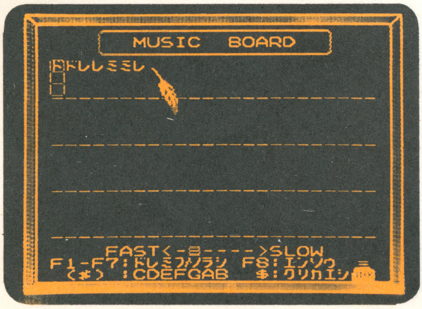
+、−、×、÷を使った4則演算はもちろん、(　)を使った複雑な計算も、式をボードに書きこんだあと、=をつければあっという間に答えを出してくれます。

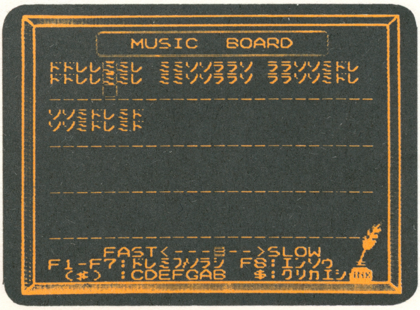
ボードはけっこう大きいのでたくさんの計算を続けて行なうことができます。ボードが計算式や答えでいっぱいになってしまっても消してまた使えます。

ミュージックボード









「コンピュータミュージック」というのがいろいろなところで話題になっています。それとは少々違いますがファミリーベーシックにも「ミュージックボード」というコンピュータが自動演奏をしてくれる便利なものが内蔵されています。

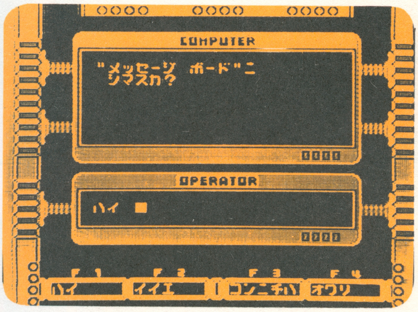
「ミュージックボード」を使ってボードに音階を書き込むと、書き込んだ順番どおりにその音階をコンピュータに演奏させることができます。

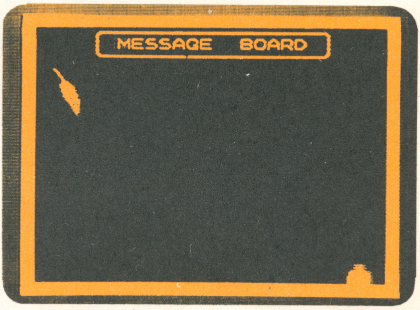
音階は3オクターブまで出せますし、さらに3つまででしたら同時に音も出すことができます。

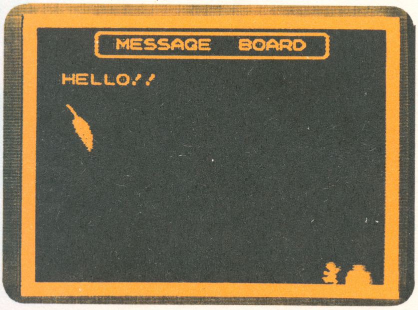
パートに分けて音階をボードに書き込めばかなり高度な演奏をさせることが可能です。

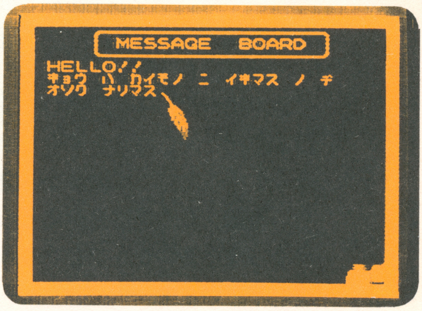
また繰り返し演奏ができますので、作った音楽を何回も聴けます。コンピュータに演奏させる速度も自由に変えられます。

メッセージボード









ファミリーコンピュータに接続したキーボードから入力された、いろいろな文字や記号などをそのまま残しておくことができます。これがファミリーベーシックに内蔵されている「メッセージボード」です。

「メッセージボード」のボードは横24文字、縦17文字の計408文字を書くことができますので、全部を使えばかなり長い文章を残しておけます。

ボードに書ける文字はアルファベット、カタカナの他に、表が作れるグラフィック文字があります。グラフィック文字を使えば、表が作れますので、こづかい帳や、料理の材料表などメモ帳代わりにボードを使うことだって可能です。

ボードにはキーボードから入力した文字をすべて表示させることができますので、タイプライターの練習用としても使えます。

この本で初めてBASICを学ばれる方へ

数年前、パーソナルコンピュータというものが世の中に出はじめたころ、ある雑誌で、「コンピュータとは何ですか？」という質問を普通の人にしてみた結果について書かれてありました。

確か一番多かった答えが「何でもできる万能な機械」とか「人間の代わりにどんな仕事もこなしてくれる人間より賢い機械」といったものでした。

今考えてみると、いささかＳＦ映画的な未来物語を想像して答えていることがとても信じられないはずです。でもパーソナルコンピュータが普及していませんから、「コンピュータ」と名のつくものに触れることができたのはごく一部の人たちだけだったのです。

コンピュータを見たことも触ったこともない人にとって、コンピュータは人工知能でありロボットでもあったのは当然だったのかもしれません。

ところがここ２、３年の間にコンピュータは「パーソナル」な形で急激に普及しはじめました。ですから今、同じ質問をしたとすれば数年前と同じ答えをする人はいないはずです。「人間が細かい指示をしてやらなくては動かない機械」とか「ソフトがないと何もできない機械」なんて答えがほとんどの人から返ってくるはずです。

よっぽど奇特な人でもその外観を何らかの形で見たことがあるはずだと思います。

確かにコンピュータは人間が細かい指示をしてやらなければ動かないし、ソフトがなければ何もできません。

でもここで一つ覚えてもらいたいことは、その「細かい指示」や「ソフト」というものを自分で作れる、ということです。

そのために、パーソナルコンピュータのほとんどが「BASIC」という言語を持っています。

私たちが未知の言語を理解できないようにコンピュータも自分の持っている言語でなければ理解することはできません。ですから、これからコンピュータを使おうと思っているのなら、コンピュータの言語を覚えなくてはならないのです。

私たちが日本語や英語を覚えたときのように、BASICを覚えるのにも時間をかけて根気よくやってください。

FAMILY COMPUTER FAMILY BASIC PERFECT BOOK

PART 2

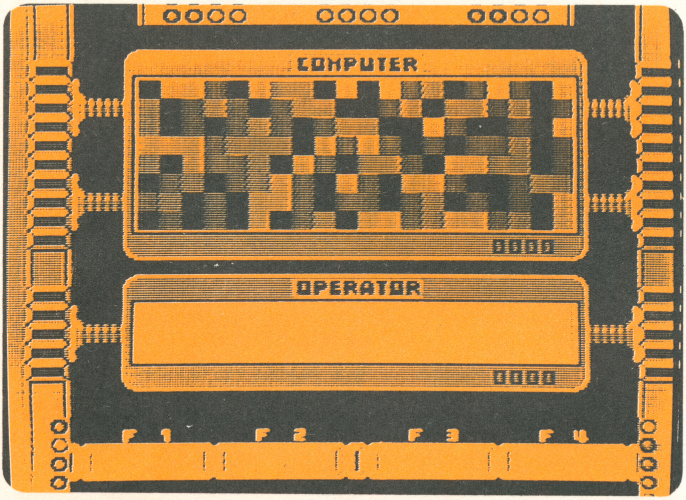
●

ファミリーベーシックの使い方

ファミリーベーシックPower ON!

ユニークなスタート。ユーモラスなベーシック画面だ

ファミリーコンピュータの電源を入れると、下のような画面がテレビに出てきます。

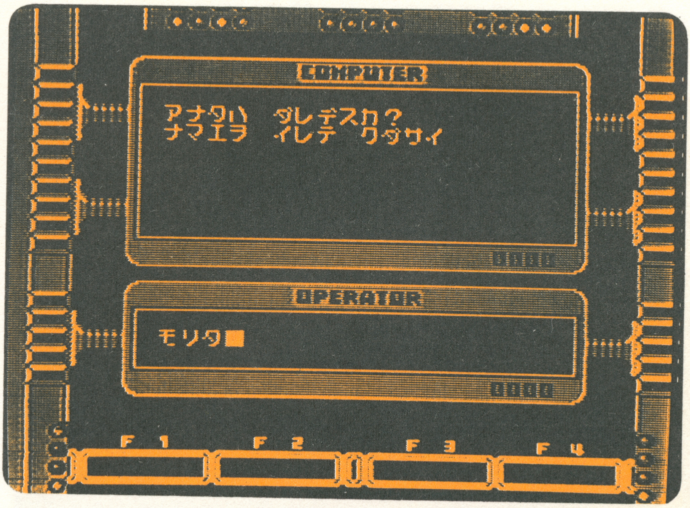


このとき、キーボードにあるキーのうち、どれか1つを押すとパネルの点滅が消えて、コンピュータが話しかけてきます。

ワタシハ
ファミリーコンピュータ　デス

ドウサヲ　カイシ　シマス

次にあなたの名前をコンピュータが聞いてきますのでキーボードから入力してください。名前は、18文字以内です。英文字でもカタカナでもかまいません。入力した名前は電源を切るまでコンピュータが記憶しています。名前の入力が終ったら RETURN キーを押します。



リョウカイ!

サギョウヲ　カイシ　シマス

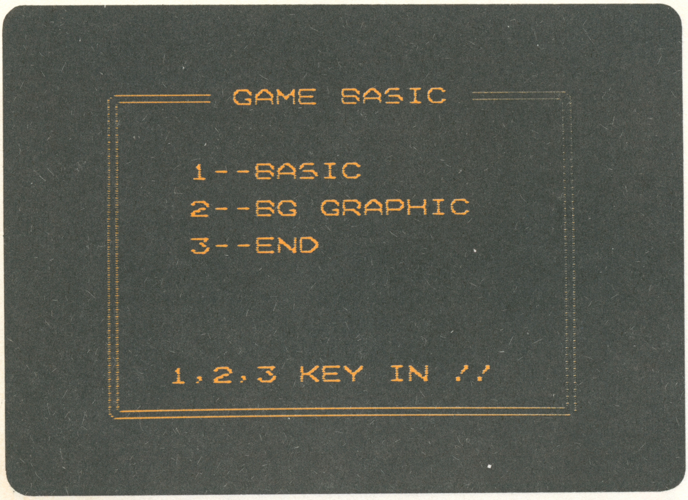
"GAME　BASIC" ニ
シマスカ?

これであなたの名前はコンピュータに確認されました。

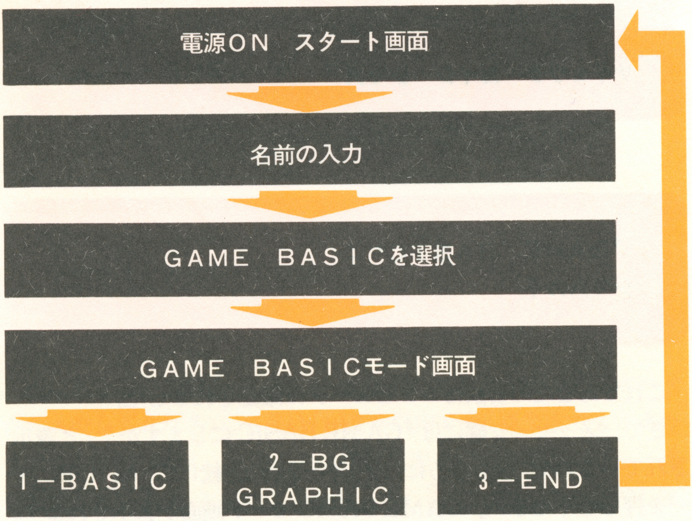
さて、メッセージ欄には「GAME BASICにしますか?」と表示されているので、おもむろに F1 キーを押し、続けて RETURN キーを押してください。GAME BASICモード画面が表示されます。

GAME BASICのメニューを選ぶ！

いくつもの機能を持つファミリーベーシック。まず基本のBASICプログラムからスタート



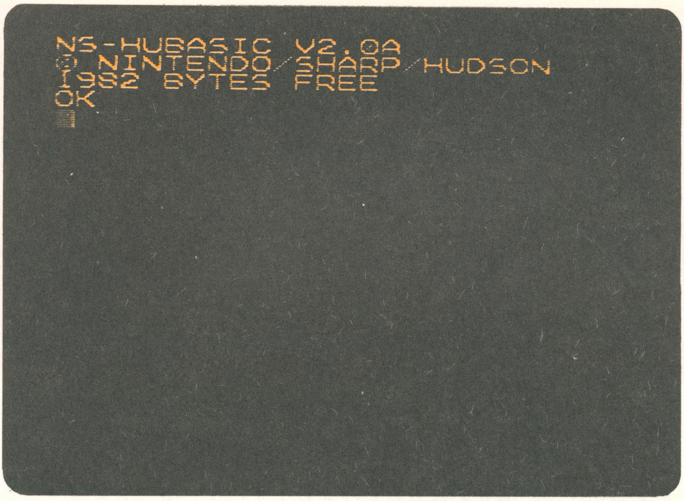
GAME BASICには2つのモードが内蔵されています。1つはプログラミングをするための「BASIC」、もう1つは背景を作るための「BG GRAPHIC」です。それぞれキーボードから、1か2のキーを押すことで選べます。3のキーを押せばスタート画面に戻ることもできます。それでは、ここまでの流れをみてみましょう。



プログラミングの前に

BASICのスタート画面が表示される。スタート直前

さてそれでは、GAME BASICモード画面で、[I]キーを押してBASICを選んで下さい。下のような画面になりましたか。



●プログラムの消去

もし、上のような画面にならないで、「OK」と表示されるだけでしたら、前に使ったときのプログラムが残っています。とりあえず、ここではNEW命令を使ってプログラムを消してください。キーボードから[N]，[E]，[W]と入力し、[RETURN]を押します。

働き	―― メモリ上にある変数，プログラムをすべて消す
文法	―― NEW

すると、画面は次のようになります。画面に表示された「OK」はコンピュータが命令を理解し、実行した、という意味です。

NEW←――キーから入力した命令
OK←――コンピュータが命令の実行を完了した、という印
■←――カーソル

いよいよプログラムの入力だ

キーボードに慣れていないと打ち込みが大変だけど、まずはカンタンなところから

これで、プログラムを入力するための準備ができました。それではさっそくプログラムを入力してみましょう。

```
10  CLS
20  SPRITE ON
30  CGSET 1,0
40  DEF MOVE(0)=SPRITE(0,3,1,2,0,0)
50  POSITION 0,0,128
60  MOVE 0
70  GOTO 60
```

このプログラムを、頭から順に入力していくわけですが、まず最初の1行を見てください。

10　CLS

まず最初にこの部分を入力してみましょう。キーボードから

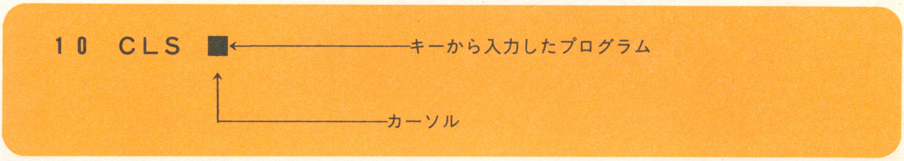
1 → 0 → スペース → C → L → S → RETURN

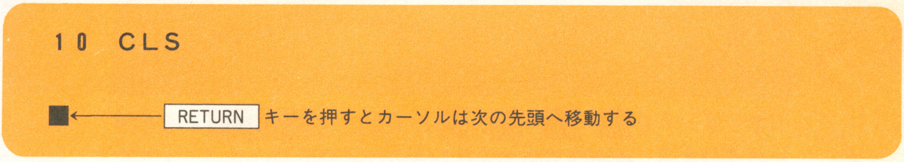
と入力します。スペース はキーボードのいちばん下にある細長い何も書かれていないキーのことです。これから先もアキがあるところは必ずこの スペース を入力してください。

また、いちばん最後は RETURN キーを必ず押さなければなりません。この RETURN キーを押すところで、コンピュータに「10行の入力が終りましたよ」ということを知らせることになるからです。



これを行なわなければ、10　ＣＬＳというプログラムは画面に表示されただけでコンピュータには記憶されません。





これからプログラムを入力するときや、プログラムの修正を行なうとき、最後に RETURN キーを押すことを絶対に忘れないでください。

なお、RETURN キーを押せば上のようにカーソルキーが自動的に次の行の先頭へ移動します。

どうです、プログラムの入力方法がわかりましたか。この要領で30ページのプログラムを全部入力してください。

入力していて、いちばん下の行までくると、いちばん上の行が画面上から消えてしまいますが、たとえ消えてしまったものがプログラムであっても、何らさしつかえはありません。画面から消えてしまっても、コンピュータのメモリ上にプログラムは残っていますので心配しないでください。

さて、プログラムの先頭に「10～60」の番号が付いていますが、BASIC ではこの番号のことを、特に行番号と呼んでいます。

行番号はこのように１０ずつでなくてかまいませんが、後でプログラムを追加したり、また見やすさの点から、１０ずつにするのが望ましいと思われます。行番号に使える数は0から６５５３５までの整数です。

また、行番号を呼ぶときは「１０行、２０行」というようにします。以後何回も出てきますので覚えていてください。

入力したプログラムの確認をしてみよう

LIST 命令で打ち込んだプログラムを見る。カンゲキ

プログラムの入力が終ったところで今度は入力したプログラムを確認してみましょう。短いプログラムなので間違いなく入力できたと思いますが、これを機会に修正方法を覚えてください。

さて、まず SHIFT キーを押しながら HOME キーを押します。すると画面が消えカーソルがいちばん左上に来たはずです。さきほど入力したプログラムを、ＬＩＳＴという命令を使って画面に表示させます。

修　正	—— メモリ上にあるプログラムを画面に表示させる
文　法	—— ＬＩＳＴ〔<行番号>　〔—<行番号>　〕
省略形	—— Ｌ.

キーボードで L 、 I 、 S 、 T と入力し、 RETURN キーを押してください。画面は下のようになったはずです。

```
LIST
10 CLS
20 SPRITE ON
30 CGSET 1,0
40 DEF MOVE(0)=SPRITE(0,3,1,2,0,0)
50 POSITION 0,0,128
60 MOVE 0
70 GOTO 60
OK
■
```

このようにＬＩＳＴは行番号を入れずに RETURN キーを押すと、入力したプログラムをすべて表示します。また、ＬＩＳＴは行番号を後に付けると、特定の行のみを画面に表示させることができます。

たとえば、ＬＩＳＴ　10とすれば10行を表示し、ＬＩＳＴ　20−40とすれば、20行から40行を表示します。長いプログラムを表示させる場合や、特定の行だけを見たいときは非常に便利です。

```
LIST 10
10 CLS
OK
LIST 20-40
20 SPRITE ON
30 CGSET 1, 0
40 DEF MOVE (0)=SPRITE (0, 3, 1, 2, 0, 0)
OK
■
```

なお、〜からの意味をなす「−」は０の隣りにあるキーでマイナスです。これの代りに「,」を使ってもかまいません。ＬＩＳＴ10−やＬＩＳＴ−40なんて使い方もできますのでやってみてください。

●画面編集（スクリーンエディット）

さて、それでは最初にＬＩＳＴで画面に出したプログラムを30ページのプログラムと比較してください。全く同じであれば問題ありませんが、違いがあればそれを直さなくてはなりません。

仮に10行が、10　ＣＬＲとなっていたとしましょう。まず最初にキーボードの右側の［←］［→］［↑］［↓］のキーを使ってカーソルを直したい場所までもっていきます。カーソルは押したキーの方向へ１つずつ動き、キーを押しっぱなしにすれば連続的に動きます。カーソルがＲの位置まできたら、［S］を押し、そのまま［RETURN］キーを押します。

これで修正は完了しました。このように画面上で修正や編集ができることをスクリーンエディット機能といいます。

この他、カーソルキーの上にある［INS］や［DEL］キーを使えば、さらに高度な編集ができます。下の図を参考にしてください。

POSTION　［INS］キーを押す
↓
POS TION

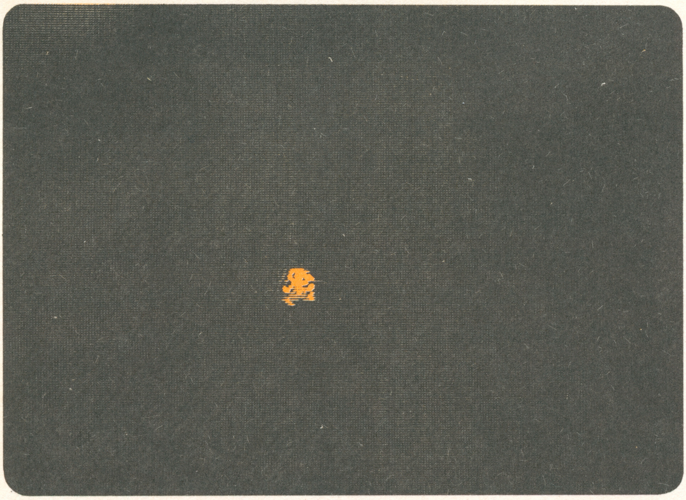
SPURITE　［DEL］キーを押す
↓
SPRITE

プログラムの実行はRUNで行なう

入力したプログラムの正誤が確認できたら、今度はそのプログラムを実行させてみましょう。プログラムを実行させるための命令はRUNです。

働き	——	プログラムを実行させる
文法	——	RUN〔<行番号>〕
省略形	——	R.

では、キーボードから R , U , N と入力し、RETURN キーを押します。すると下のような画面になるはずです。



実行してみてわかったと思いますが、これは、マリオを左から右へ動かすプログラムです。もし、実行させてもこのような画面にならず、ピーッと音がして

? SN ERROR IN △△

と表示されたら△△行に間違いがあります。LISTを使ってもう一度画面にプログラムを表示させ、入力の間違いを直して再びプログラムを実行させてみてください。

上のような画面になるまでは、どこかに入力間違いがありますので根気よくさがしてみましょう。

プログラムを中断するには

このプログラムを実行させていて気付いたと思いますが、実行させた後、いつまでたっても終りません。それもそのはずです。このプログラムは終らないように作ってあるからです。

では、このプログラムを止めるにはどうしたらいいでしょうか。電源を切ったり、リセットスイッチを押すのも１つの方法ですが、それではせっかく入力したプログラムが消えてしまいます。

プログラムを消さずに止めるのには STOP キーを使います。ためしに STOP キーを押してみてください。

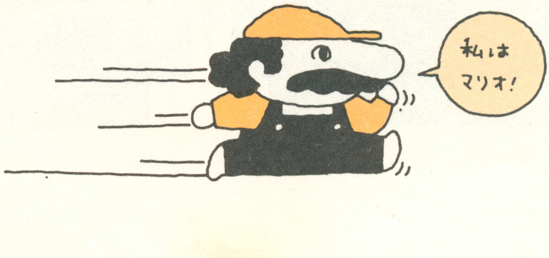
```
BREAK IN 70
OK.
■
```

画面にはこのように表示されました。これは70行でプログラムが中断されました、という意味です。場合によってはこれがＢＲＥＡＫ　ＩＮ 60となりますが別におかしくありません。このプログラムが60行と70行を繰り返し実行しているから偶然そうなったのでしょう。

STOP キーはこのプログラムのように終りがないものを止めることができますが、もちろん終りのあるプログラムを途中で止めることだってできます。STOP キーを使ってプログラムを止めてもＬＩＳＴでプログラムを画面に表示させることができるし、ＲＵＮで再び実行させることもできます。

ためしに再び R, U, N と入力し、RETURN キーを押してみてください。再びマリオが画面上に左から右へ動き出しましたね。

この本に載っているプログラムや自分で作ったプログラムが正常に動かなくても何もあわてることはありません。STOP キーを押せばいつでもプログラムは止まります。



プログラムを保存する2つの方法

プログラムの保存（セーブ）する方法をしっかり覚えよう

プログラムを入力して実行させたのはいいけど、このまま電源を切ってしまえば、プログラムは消えてしまいます。短いプログラムだったらまたキーボードから入力すればいいのですが、これが長いプログラムだったらそうもいかないでしょう。そこで、プログラムの保存が必要となるわけです。

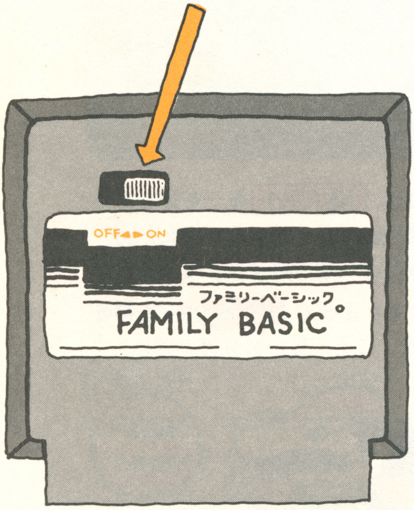
さて、ファミリーベーシックではプログラムを保存する方法が２つあります。１つはバックアップスイッチを入れる、もう１つはカセットテープに記録することです。

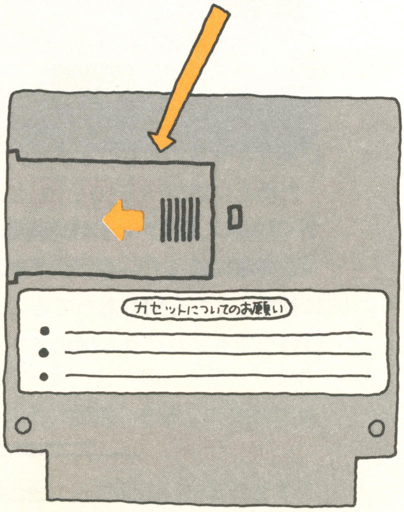
それでは、この２つの方法の手順を見ていきましょう。

●便利なバックアップ機能

ファミリーベーシックのカセットには下のようなスイッチが付いています。このスイッチをONにすれば今入力したプログラムを保存することができます。しかし、電池が入っていなかったり、スイッチを入れるタイミングを間違えるとONにしたにもかかわらず、プログラムは消えてしまいます。必ず手順どおりにやってください。

表　バックアップスイッチ　　　　　　　裏　電池

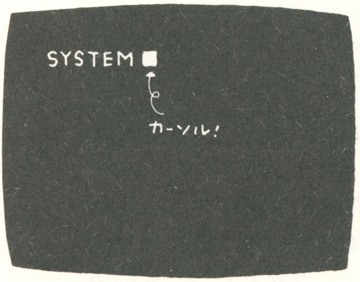




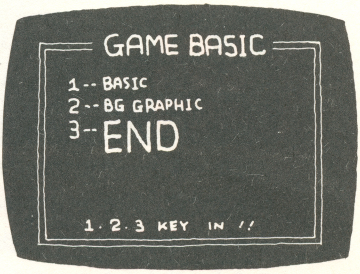
まず、LISTを使ってプログラムがメモリ上に残っているかを確かめます。次に、キーボードから S, Y, S, T, E, M と入力し、RETURN キーを押します。するとGAME BASICモード画面になりますので、3のENDを選びます。これは 3 のキーを押せばいいのです。キーが押されると画面はさらにスタート画面へと変わります。このとき、バックアップスイッチをONにしてください。

これでプログラムの保存ができました。後は電源を切っても、カセットをファミリーコンピュータからぬいてしまってもかまいません。電池がなくなるまでプログラムは保存されます。

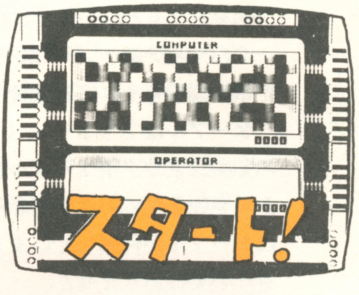
よくわからなかった人は、下の図を参考にしてみてください。



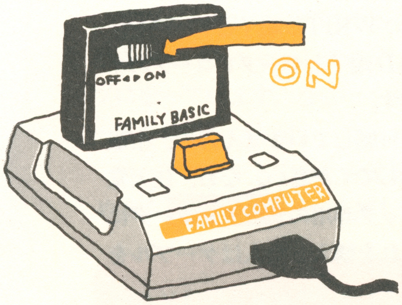
キーボードからSYSTEMと入力し、RETURN キーを押す



GAME BASICモード画面で3のENDを選ぶ（3 のキーを押す）



スタート画面が表示される



バックアップスイッチをONにする

●カセットテープに保存するには

バックアップスイッチは手間がかからずプログラムを保存することができますが、１つのプログラムしか保存できません。しかも、バックアップスイッチを使ってプログラムを保存しておいても、カセットの電池がなくなってしまえば、せっかくのプログラムが消えてしまいます。

また、再び電源を入れた後、GAME BASIC 以外のボードなんかを使ったりしてもプログラムは消えてしまいます。

それじゃあ困る、と思った人や、いくつものプログラムを保存したい人は、ＳＡＶＥを使って、カセットテープに記録してください。

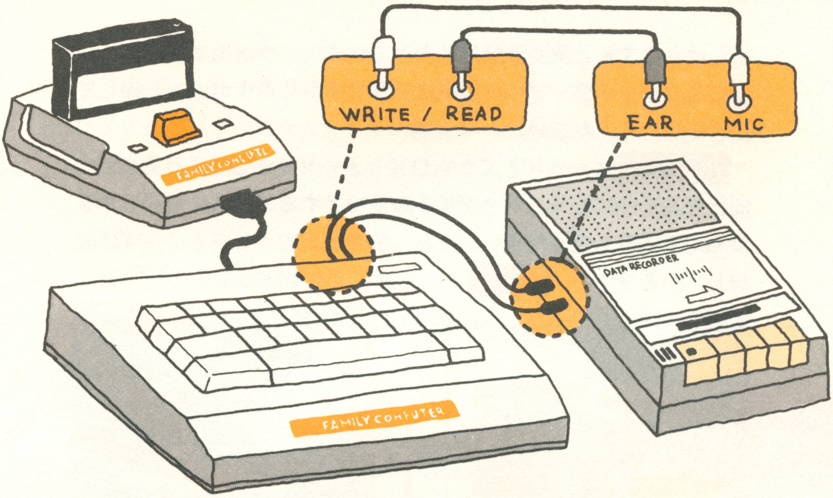
働　き	――	プログラムをカセットテープに記録する
文　法	――	ＳＡＶＥ〔"ファイル名"〕
省略形	――	ＳＡ.

カセットテープにプログラムを記録するには、まずテープレコーダが必要です。パソコン用データレコーダはむしろ不適でモノラルのオーディオ用が向いていますが、できればこちらの方を使いたいものです。適当なものが手もとになかったら、ファミリーベーシック専用にデータレコーダを１台購入してください。

それではさっそくプログラムをカセットテープへ記録してみましょう。まず最初に次の図のようにキーボードの後ろにある端子と、データレコーダの端子をケーブルで接続してください。

図を見てもわかるようにキーボードの WRITE 端子とデータレコーダのマイク端子、キーボードの READ 端子とデータレコーダのイヤホン端子をそれぞれ接続するわけです。

ケーブルはファミリーベーシックに付属していませんので、市販のミニジャック用ケーブルを２本購入してください。なお、任天堂から発売されているファミリーベーシック専用のデータレコーダにはケーブルが付属していますので、これを購入した人はケーブルを買う必要はありません。



操作のしかたと保存のためのプログラム

まず最初にＬＩＳＴでプログラムがあるかどうかを確認します。プログラムがメモリ上に残っていたら、キーボードからＳＡＶＥ"ＳＡＭＰＬＥ１"と入力します。" は SHIFT キーを押しながら 2 のキーを押すことで出せます。このとき、まだ RETURN キーを押しません。

```
SAVE "SAMPLE1" ■
```

次にデータレコーダにカセットテープをセットします。一般に使用されているカセットテープは最初の方が録音できない透明のテープになっています。なるべくなら、頭から録音できるパソコン用テープを使用した方がいいでしょう。

テープがセットできたら録音ボタンを押します。テープが回りはじめたら RETURN キーを押してください。画面は次のようになります。

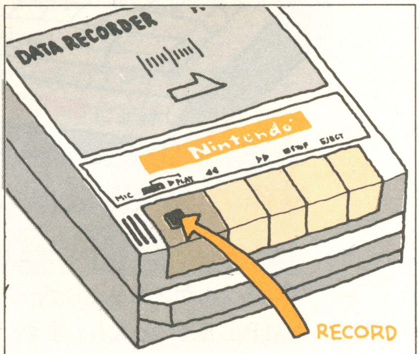
```
SAVE "SAMPLE1"
WRITING SAMPLE1
```

しばらくすると画面にＯＫと表示されカーソルが出ます。これが出たらカセットテープへの記録は終りですから停止ボタンを押して録音をやめてください。

RETURN キーを押してからＯＫとカーソルが表示されるまでの間に録音をやめたり、ケーブルをぬくなどすると正常に記録できませんので注意してください。なお、ＳＡＶＥでカセットテープに記録している途中で、STOP キーを押しても無効となります。

```
SAVE "SAMPLE1" ■
```

ＳＡＶＥ〝ＳＡＭＰＬＥ１〟と
キーボードから入力する



データレコーダの録音ボタンを
押す

```
SAVE "SAMPLE1"
WRITING SAMPLE1
```

RETURN キーを押す

```
SAVE "SAMPLE1"
WRITING SAMPLE1
OK
■
```

となったらデータレコーダを
停止させる

セーブしたカセットテープを確かめるには

これでカセットテープにプログラムが記録できた、と思いがちですが、実はまだわからないのです。確かにテープは回っていたかもしれませんが、プログラムとしての録音レベルを満足している、という保証はどこにもありません。せっかく入力したプログラムなんですから、また入力しなおすなんてことがないようにしたいものです。

そこで、このLOAD？を使ってカセットテープにプログラムが記録できているかを確かめてください。

働き ——	テープに記録したプログラムがメモリ上のプログラムと同じかを確かめる
文法 ——	LOAD？〔"ファイル名"〕
省略形 ——	LO.？　またはLO．P.

まず、カセットテープをプログラムを記録した頭の部分まで巻きもどします。次にデータレコーダのトーンコントロールをいちばん高い方に、ボリュームレベルを最大より少し手前にセットします。

そして、キーボードからLOAD？"SAMPLE 1"と入力し RETURN キーを押します。あとは、データレコーダの再生ボタンを押せばOKです。

さて、プログラムがカセットテープに正しく記録されている場合画面は下のようになります。

```
LOAD? "SAMPLE1"
LOADING  SAMPLE1
OK
■
```

いちばん最初からこのようになれば問題ないのですが、なかなかこううまくはいきません。

もし、LOADING～の表示がいつまでたっても出てこなかったら、ボリュームレベルが小さすぎます。STOP キーを押し、もう一度ボリュームレベルを大きくしてからやり直してください。

それでも何も表示されないようでしたら、カセットテープにプログラムの頭の部分が記録されていないか、ケーブルの接続の間違いで何も記録されていないかのどちらかです。39ページを見て、ケー

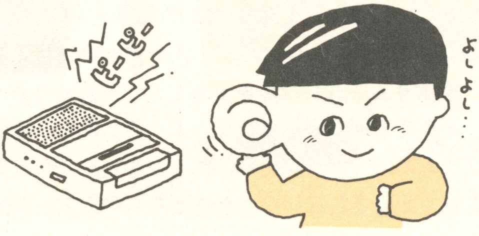
ブルの接続を確かめてみてください。さて、LOADING～の表示が出ても、下のように？TP　ERRORの表示が出てしまうこともあります。

```
LOAD?  "SAMPLE1"
LOADING  SAMPLE1
?TP  ERROR
OK
■
```

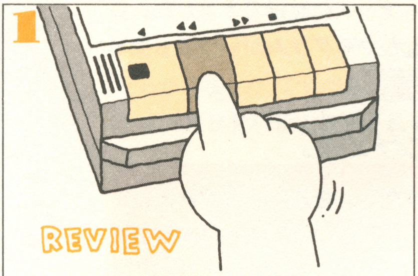
このような場合、いろいろな原因が考えられますが、とりあえずボリュームコントロールを回して、大きくしてみます。それでもなお同じことが起こるようでしたら、今度は小さくしてみてください。なお同じ結果でしたら、テープの不良や他の何らかの理由で途中の部分に記録されていないところがあります。カセットテープを交換し、再びSAVEしてみてください。

こんなふうに言っていると、カセットテープへプログラムを記録することは非常にめんどうだと思われるかもしれません。しかし、このボリュームコントロールの調整も、ひとたび適当な位置をみつけてしまえば、その次からはほとんどさわらなくても大丈夫です。普通のテープレコーダは、イヤホン端子にプラグを差し込むとスピーカーから音が出ませんが、パソコン専用のデータレコーダならほとんどのものがプラグをさしてあってもスピーカーから音が聴けるようになっています。

音を聴きながらでしたら、手さぐりでやっているよりもいくぶん安心できるはずです。

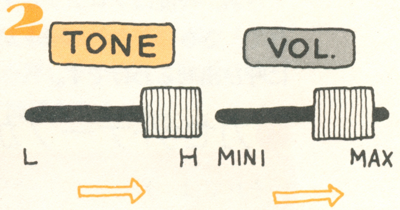


1



カセットテープをプログラムを記録した頭の部分まで巻きもどす

2



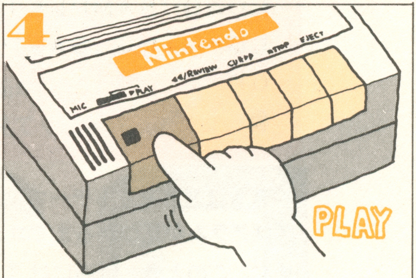
データレコーダのトーンをいちばん高く、音量を最大より少し小さめにセッティングする

3

```
LOAD? "SAMPLE1"
```

LOAD? "SAMPLE1"
と入力し RETURN キーを押す

4



データレコーダの再生ボタンを押す

5

```
LOAD? "SAMPLE1"
LOADING SAMPLE1
OK
■
```

LOADING～とOKが表示されたら完了！

6

```
?TP ERROR
OK
■
```

?TP ERRORの表示が出たら音量の位置を前回と変えて、もう1度最初から

プログラムを読み込む方法

正しく保存されたプログラムを読み込んでみよう

さあ、今度はプログラムをカセットテープに記録できました。これで、さきほどキーボードから入力したプログラムがいつでも簡単に呼び出せるわけです。

●カセットテープから読み込む

それでは、カセットテープからプログラムを読み込んでみましょう。カセットテープからプログラムを読み込むのにはLOADを使います。

働き	—— カセットテープからプログラムを読み込む
文法	—— LOAD〔"ファイル名"〕
省略形	—— LO.

まず、カセットテープを読み込みたいプログラムの入っている頭の部分まで巻きもどしてください。次に、データレコーダをトーンコントロールとボリュームレベルをLOAD？でエラーを出さずに読み込んだ位置に設定します。

そして、キーボードからLOAD "SAMPLE 1" と入力し、RETURN キーを押してください。" は SHIFT キーを押しながら 2 キーを押して出します。

これであとは、データレコーダの再生ボタンを押せばいいのです。もし、さきほど記録したプログラムがコンピュータに読み込めたら画面は次のようになります。

```
LOAD "SAMPLE1"
LOADING  SAMPLE1
OK
■
```

さきほどカセットテープに記録したプログラムが先頭にない場合プログラムが見つかるまでコンピュータはプログラムを読み込みません。このとき、画面には " " でくくったものと違うものがSKIP～の形で表示されます。コンピュータが、カセットテープに指定されたプログラムと違うものが入っています、と教えてくれている

んですね。

　ここでこのカセットテープがさきほど記録したものと違うものだとわかりました。でしたら STOP キーを押してください。画面には、?TP　ERRORと表示されますが、関係ありません。

```
LOAD "SAMPLE1"
SKIP  GAME1
SKIP  GAME2
?TP  ERROR
OK
■
```

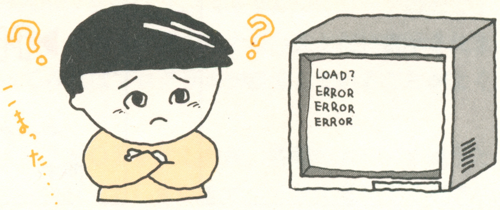
　LOADを使ってプログラムを読み込むときに、LOAD?で正しくテープに書き込まれていることや、ボリュームレベルをどのくらいにすればいいかを確認してあれば、まずプログラムがうまく読み込めないことはありません。

　でも、もしプログラムがうまく読み込めなかったり、?TP　ERRORが画面に表示されたら、LOAD?のときと同じような手順でボリュームレベルを調整してみてください。それでも読み込めない場合、ほかになすすべはありません。他のデータレコーダを使えば読み込むことができるときもありますが、ほとんど無理だと思ってください。

　また、LOADはSAVEするときに付けた " でかこまれた名前を省略すると、カセットテープを再生したときに、いちばん最初に出てきたプログラムをコンピュータに読み込みます。

　次のページにロードの手順が図で描かれています。

　よくわからなかった人はそちらを見てください。



1



カセットテープをコンピュータに読み込ませたプログラムの入っている巻頭の部分まで巻きもどしておく

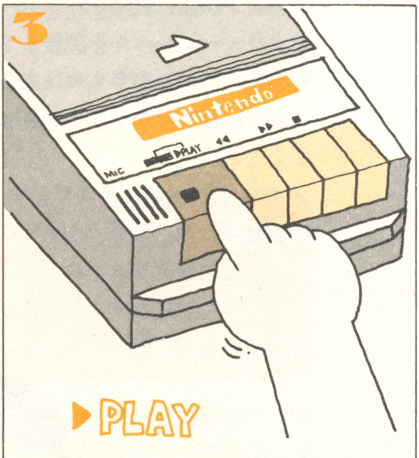
2

```
LOAD "SAMPLE1"
```

キーボードからLOAD"SAMPLE1"と入力し、RETURNキーを押す

"はSHIFTキーを押しながら2のキーを押して入力する

3



データレコーダの再生ボタンを押す

4

```
LOAD "SAMPLE1"
LOADING  SAMPLE1
OK
■
```

LOADING～が表示され、OKとカーソルが出たらプログラムの読み込みは完了

●バッテリー・バックアップした プログラムを読み込むには……

プログラムを保存する方法にはカセットテープに記録する方法の他に、バックアップスイッチを使うというのがありました。これはファミリーベーシックのカセットに入っている電池を利用し、プログラムを保存する、というものでしたね。

それでは、このバックアップスイッチを利用したとき、どのようにすればプログラムが復元するのかみてみましょう。

①ファミリーコンピュータの電源を入れる

↓

②スタート画面が表示されるので何かキーを押す

↓

③「バックアップスイッチをＯＦＦにしてください」と表示される

↓

④バックアップスイッチをＯＦＦ側にする

↓

⑤「ＢＡＳＩＣのデータが残っています」と表示される

これでGAME BASICを選び、GAME BASICモード画面で1のBASICを選びます。すると画面の左上にＯＫとだけ表示されます。ＬＩＳＴをやってみればプログラムが復元していることがよくわかるでしょう。

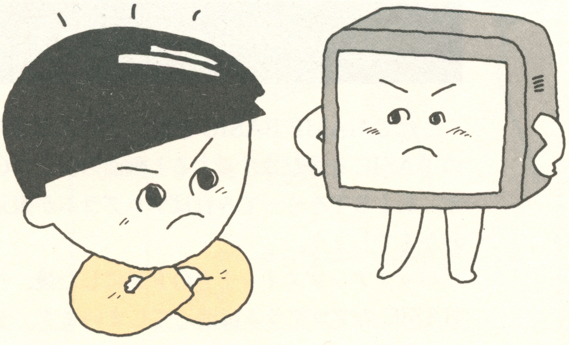
バックアップスイッチをＯＦＦにした後、各種のボードを使うとBASICのプログラムは消えてしまいます。十分注意してください。

便利なコントロールコードについて

画面上でプログラムを修正するには INS や DEL キーを使うことはもうわかったと思いますが、この他にも便利な機能がたくさんあります。

たとえば CTR キーを押したまま E キーを押すとカーソル以降の1行分を消すことができます。ＬＩＳＴでプログラムを表示させた後、特定の行の途中から最後までカーソルを消去したい場合は、その場所にカーソルを持って来て、 CTR ＋ E を押せばいいのです。この後で RETURN キーを押すこともわすれないでください。

CTR+	A	INS モードのON／OFFスイッチ
	C	BREAK。プログラム実行中はきかない。
	D	初期設定。 CGEN2、SPRITE OFF、CTR＋Aの解除。カラーパレットをスプライト、バックグラウンド共に、バックグラウンド用のパレットコード1にする
	E	カーソル以降1行分の消去
	G	BEEP音を出す
	H	DEL と同機能
	J	行送りをする
	K	カーソルをホーム位置に戻す
	L	画面のクリア
	M	1行入力して改行
	R	INS と同機能
	V	カナモードにする
	W	英・数字状態にする
	Z	カーソル以降画面下までのクリア



FAMILY COMPUTER
FAMILY BASIC
PERFECT BOOK

PART 3

●

ファミリーベーシック基本編

マニュアルをうまく使おう

いってみればマニュアルは教科書。基本はしっかり覚えよう

ファミリーベーシックは一見するとオモチャっぽいですが、やはりちゃんとしたコンピュータです。BASIC 言語を使ってプログラムを組もうとする場合に、他のパーソナルコンピュータより簡単にできる、なんてことはありません。

すでにいろいろなコンピュータを使っていて BASIC でプログラムを組むことに慣れている人は問題ありませんが、そうでない人は、まず取扱説明書の見方から覚えましょう。

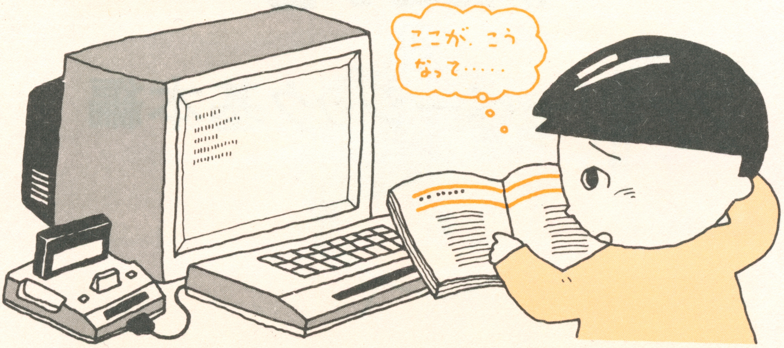
●マニュアルは何のためにある?

テレビや冷蔵庫などの電化製品には、必ずマニュアル(取扱説明書)が付いています。どんな電化製品であっても、最初はその使い方がわからないわけですから、当然このマニュアルを見なくてはなりません。

しかし、多くの電化製品の場合、ひとたびその使用方法がわかってしまえば、もう二度とマニュアルを見る必要はなくなってしまうはずです。これは、これらの電化製品の使い方が簡単で、しかも用途が限られているからなんですね。

では、コンピュータの場合はどうでしょうか。どのコンピュータも基本的な「使い方」はちょっとマニュアルを読めばすぐに理解できます。でも、だからといってマニュアルがいらなくなる、ということはありません。

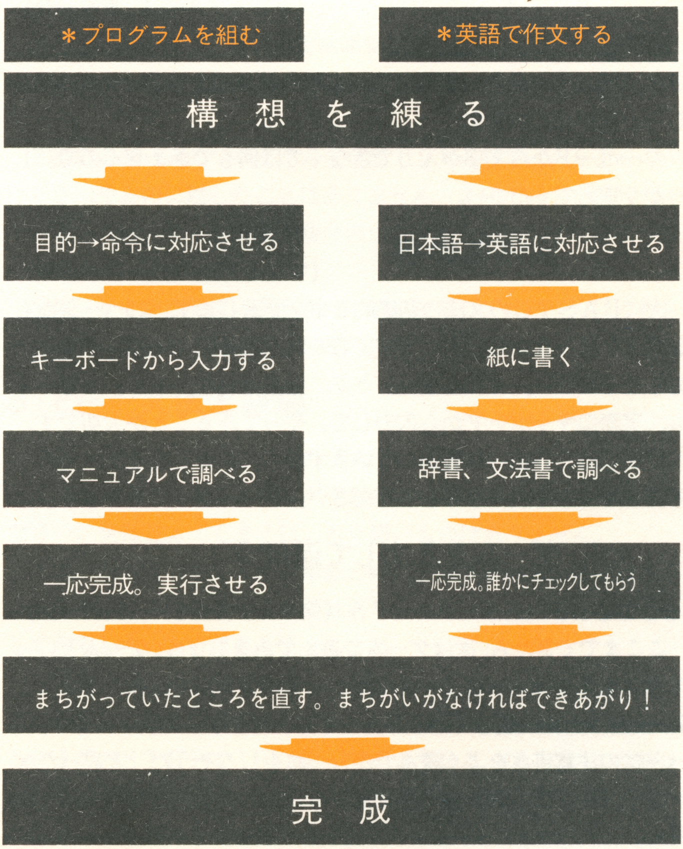
コンピュータの場合は、「使い方」の次に「自分でプログラムを組む」ことを覚えなければならないからです。



「自分でプログラムを組む」ということは、英語で作文をすることによく似ています。ここで、はじめて英語で作文したときのことを思い出してみてください。何が必要でしたか？　そう、いちばん必要だったものは「辞書」ですね。その次は「文法書」でしょう。

これらのものがなくては、英語で作文することはできなかったはずです。

プログラムを組むとき、同じように「辞書」や「文法書」となるものが必要となります。そう、それがマニュアルなんですね。



マニュアル使用法4つのステップ

英語を覚えるのに学習の方法があるように、プログラムの作り方を覚えるのにも学習の方法があります。英語の辞書を頭から全部覚えても文法を理解していなくては無意味です。同じように、BASICの命令を全部覚えても、文法がわかっていなくてはやはり無意味なことになってしまうのです。

そこで「辞書」であり「文法書」でもあるマニュアルのうまい使い方を紹介しましょう。もちろん、自分が理解できていることは飛ばしてしまってかまいません。

それではＳＴＥＰ１から初めてください。

STEP1:文字の入力

(マニュアルの4，5ページ)

まず最初は文字の入力です。マニュアルの4，5ページを見てください。英文字、数字、カナ文字、記号の入力方法やカーソルの動かし方が書かれています。

キーボードから文字が入力できなくては何もできませんので、しっかり覚えてください。

STEP2:文法基礎編

(マニュアルの15〜48ページ)

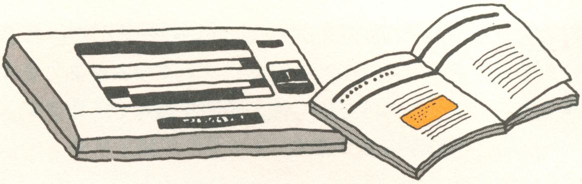
次にＢＡＳＩＣの文法の基礎的な事を学習します。16ページから説明文を読みながら、太い文字で書かれた部分をキーボードから入力してください。くだらない、とか、結果はわかってる、と思っても、実際にキーボードから入力して、やってみることが大切です。

説明文はていねいに書かれていますから、わからなくなってもなげ出さず、よく読みなおしてください。48ページまであります。

STEP3:ゲームを入力して遊ぼう

(マニュアルの91〜101ページ)

ここまでできたら、次はゲームで遊んでみましょう。ゲームといってもファミリーコンピュータのカセットのゲームではありませんよ。91〜101ページに載っているＢＡＳＩＣプログラムのゲームです。92ページの注意書きをよく読んでから、おもしろそうだ、と思ったゲームのプログラムを、キーボードから入力して実行させてみてください。



入力間違いがなければゲームができますが、たいてい間違いがありますので、何らかのエラーが表示されるか、正常に動きません。入力したゲームが正しく動くまでマニュアルと画面を見くらべ、間違いを直してください。絶対に途中であきらめてはいけません。間違いをさがすことで、命令の役割やプログラムの流れがつかめてくるからです。

また、役割りのよくわからない命令が出てきたら、110ページのアルファベット順命令語索引でマニュアルを引いてみることも大切です。

7本あるゲームのプログラムを全部入力する必要はありませんが、たくさんやるのに越したことはありません。キーボードを何回もたたいていれば、キーの位置や命令のつづりも自然に覚えてしまうからです。

なお、せっかく間違いを直したのですから、プログラムはカセットテープに記録しておいてください。

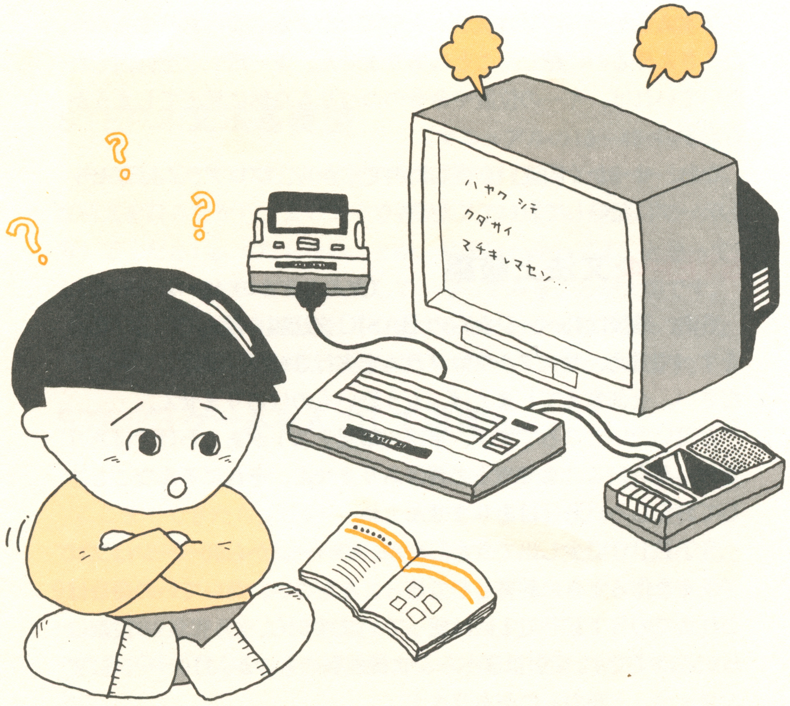
STEP4:文法上級編 (マニュアルの 49〜90ページ)

さあ、最後は49ページからのBASIC文法編を読んでいき、命令をすべて覚え、サンプルプログラムを実行させて、確認してみてください。全部覚えてしまえば、もう恐いものはありません。

と言いたいところですが、単語の暗記テストがあるわけではありませんから、ここまでやる必要はありません。それにそんなことをしておもしろいわけはありませんよね。

ＳＴＥＰ３で少し出てきましたが、49ページから90ページはプログラムを作るときに辞書として使ってください。たいていの命令はこれまでのＳＴＥＰで１回は使っていますから、だいたいの役割はわかるはずです。索引は50ページに機能別のものが、110ページにアルファベット順のものがあります。

それでは、これからBASICの文法の基礎を学んでいきましょう。ここでは、それぞれの項目の最初にプログラムがありますので、それを入力し、実行してみてから後の説明を読んでください。そうすればいっそう理解が深まります。



画面表示と四則演算

足し算・引き算・かけ算・割り算。これゲーム作りには大切なのです

```
10 CLS
20 PRINT "2+3";2+3
30 PRINT "2-3";2-3
40 PRINT "2*3";2*3
50 PRINT "2/3";2/3
60 END
```

●BASICの四則演算

BASICで加減乗除を行なうための記号は一般に使われているものと乗算、除算のみ違います。乗算は×の代りに*、除算は÷の代りに／を使います。

さて、このプログラムを実行させてみると、結果は次のようになります。

```
2+3= 5
2-3=-1
2*3= 6
2/3= 0
```

おや？　除算だけ答えがおかしいですね。本当なら0.6666……というのが答えのはずなんですが…………。

そう、実はこのBASICでは小数が使えないのです。だから、小数になってしまう2／3の答えが0になってしまったんですね。プログラムを作っていると、知らず知らずのうちに小数を使ってしまうことがよくあります。エラーの原因となりますので、注意してください。

●PRINTは画面表示の命令

プログラムを実行させてみてわかったと思いますが、PRINTというのが、画面表示をさせるための命令です。

まず、プログラムの20行を見てください。PRINTの後に2+

3が2つあります。ところが、実行結果では片方の2＋3はそのまま表示されていて、もう片方は2＋3を計算した結果が表示されています。PRINTは〝 〟でくくられたものはそのまま表示するわけです。

PRINTでの ；と ，の役割

このプログラムには「；」が4つ入っています。これは一見すると何の意味も持っていないようですが、実は画面表示と密接な関係があります。

ためしに、さきほどのプログラムにある「；」を全部「,」に変えてみてください。そしてプログラムを実行させると次のようになるはずです。

```
2＋3＝          5
2－3＝         －1
2＊3＝          6
2／3＝          0
```

これでもうわかったと思いますが、「；」を使えば間隔を取らずに表示を行ない、「,」を使えば適当な間隔を取って表示するわけなんですね。

このプログラムでは、そのまま表示するものと計算して表示するものの間に使っていますが、PRINTで表示させるものの、いちばん最後に付けてもかまいません。

ためしに、このプログラムをさきほど「；」を「,」にしたところを再び「；」にしてから、20行から50行の「3」の後に「；」を付けてみてください。プログラムを実行させると1行で表示されます。



数値変数と代入について

数字のようで数字じゃない。BASICは計算機能も充実している

```
10 CLS
20 A=25
30 B=12
40 PRINT "A=";A
50 PRINT "B=";B
60 PRINT "A+B=";A+B
70 PRINT "A-B=";A-B
80 PRINT "A*B=";A*B
90 PRINT "A/B=";A/B
100 END
```

●数値変数

上のプログラムを実行させると、結果は次のようになります。

```
A= 25
B= 12
A+B= 37
A-B= 13
A*B= 300
A/B= 2
```

なにか、連立方程式を答えから作ってしまったような感じですね。でも、この結果からAを25，Bを12としたとき、4つの方程式を計算していることはすぐにわかります。

BASICでは、このAやBのように、数を入れられる英文字を特に変数と呼んでいます。変数はAやB以外の英文も使えますし、AXやDYのように2文字を組み合わせて使うこともできます。

3文字以上英文字を使った変数も作れますが、最初の2文字が同じだと違うものにはなりません。つまり、TATEという変数とTAKASAという変数を使った場合、これら2つの変数は、コンピュータにとっては同じものに見えてしまうわけなんです。

●＝(イコール)による代入

BASICでは＝（イコール、等号）に「等しい」という意味以外に「代入する」という意味も持っています。たとえば、

Ａ＝１０　　　　・・・・①
Ａ＝Ａ＋１２　　・・・・②

という２つの式があるとします。数学的に考えた場合、この２つの式を満足させるAの値はありませんが、これをＢＡＳＩＣで①、②の順に実行させた場合、Aの値は22になります。

つまり、①で10を代入し、②でそのAに12を加えた値、22を再びAに代入しているわけですね。

「＝」の「右辺を左辺に代入する」という働きはひんぱんに使われますので覚えておいてください。

●INPUTによる代入

このプログラムは、Aが25，Bが12のときの計算結果を表示しますが、10行と20行の数値の部分を変更すれば、他の値のときの計算結果も表示できます。

しかし、いちいちプログラムを直して実行させるのは非常に面倒です。そこで、このAとBの値を、プログラム中でたずねてくるようにしましょう。前に入力したプログラムを消さないで、次のプログラムを入力してください。

```
20  INPUT "A=", A
30  INPUT "B=", B
```

さて、入力が終わったら、実行してみてください。画面の左上に次のように表示されます。

```
A=■
```

とりあえず33と入力し、RETURN キーを押します。するとBもAのときと同じように表示されますので、97と入力し RETURN キ

ーを押します。画面には次のように表示されます。

```
A=  33
B=  97
A+B=  130
A-B=  64
A*B=  3201
A/B=  0
```

これで、さきほどの4つの式をAが33，Bが97のときについて計算していることがわかりました。ＩＮＰＵＴというのは、キーボードの入力を変数に取り込むことができる命令なんですね。

10行と20行のプログラムを見てもわかるように、ＩＮＰＵＴもＰＲＩＮＴと同じく、〝〞でかこんだものをそのまま画面に表示することができます。

さて、10行と20行のＩＮＰＵＴの後にある「,」を「；」に変えてみてプログラムを実行させてみてください。

```
A=?■
```

すると、このように「？」マークが出たはずです。ＩＮＰＵＴでは「；」と「,」が「？」を表示するかしないかのスイッチの役割をしています。必要に応じて使いわけてください。

ここで、ためしにAに40000，Bに1を入力してみてください。計算の結果は全くデタラメなものになったはずです。これは、このＢＡＳＩＣが、－32768から＋32767までの整数値しか扱えないためです。

したがってこれを越える数値を計算させた場合、このようにデタラメな結果が出るか、？ＯＶ　ＥＲＲＯＲが出ます。代入する数値や計算結果は、必ず下の条件をみたすものにしてください。

このＢＡＳＩＣで扱える数値の条件：
－32768から＋32767までの整数

プログラムを繰り返すには?

プログラムの一部、あるいは全部を繰り返すことができる。便利だ

```
10 CLS
20 INPUT "A=";A
30 PRINT "A=";A
40 PRINT "A*A=";A*A
50 PRINT "A*A*A=";A*A*A
60 PRINT
70 GOTO 20
```

●GOTOによる繰り返し

このプログラムを実行させるとAの値を聞いてきますので、−31から+31までの間の適当な数をキーボードから入力し、RETURNキーを押してください。入力した数の2乗、3乗を計算し、表示します。

たとえばAに16を入力した場合、その結果は次のようになります。

```
A=16
A*A=256
A*A*A=4096

A=■
```

おや? 計算結果が表示されたのに、またAの値を聞いてきましたね。それでは、また何か適当な値をキーボードから入力してみてください。

すると、計算結果を表示しますが、再びAの値を聞いてきます。もう気づいたと思いますが、これは前にも出てきた「終わりのないプログラム」なんですね。

前に覚えたとおり、このプログラムを止めるには、STOPキーを押せばいいのですが、STOPキーを押さなければ終わらない、というのはあまりいいプログラムとはいえません。

この「終わらない」原因を作っているのは、70行にあるGOTO

20という命令のためです。

GOTO～というのは、後に続く行番号へプログラムの流れを変えることができるものです。この場合、GOTO　20となっていますから、70行まで実行するとまた20行を実行するわけです。

GOTO～はプログラムの流れを変えるのに重要な役割をしますが、うまく使わないとこのようにプログラムが終わらなくなってしまったり、プログラムの流れがよくわからなくなってしまうことになります。

●FOR～TO～，NEXTによる繰り返し

```
20  FOR A=1  TO  20
70  NEXT
80  END
```

60ページで入力したプログラムを消さずに、上のプログラムを入力してください。

そして実行させると画面には、上から下へパラパラパラと何やら表示されて、最後は次のようになりました。

```
A=20
A*A=400
A*A*A=8000
OK
■
```

画面の上の方を見るとAが16のときや17のときの計算結果も表示されています。どうやら画面上から消えてしまったものも、何か計算が行なわれていたようです。

実はFOR～TO～，NEXTという命令によって、Aの値が1から20まで連続的に変化させられていて、その計算結果が画面に表示されていたわけです。

さきほどのプログラムでは実行の結果がよくわかりませんから、次

のプログラムをさらに入力してみてください。

```
65 PAUSE
```

プログラムを実行させると、次のようにAが1のときの計算結果を表示し、カーソルが出ました。

```
A=1
A*A=1
A*A*A=1

■
```

このとき、何かキーを押してみてください。Aが2のときの計算結果が表示され、またカーソルが出るはずです。

これもわかるように、PAUSEという命令はキーボードから何かが入力されるまで、プログラムの実行を1時中断するものなのです。それはさておき、キーを押していくと、最初のAが1のときからAが20のときまで、順に1ずつ増えた値について計算結果を表示していきますね。つまり、FOR A=1 TO 20は「Aの値を1から20までで1ずつ増しなさい」という命令なのです。

ただし、FOR～TO～はNEXTがないと値を増すことはできません。逆にいえば、NEXTがあるまでは値を増さないわけです。

また、Aが20になってからもNEXTがあるわけですから、繰り返しが終わった後、Aの値は1増えた21になります。ためしにプログラムの実行が終わってOKが表示された後、PRINT Aを実行させてみてください。

```
20 FOR A=1 TO 20 STEP 2
```

FOR～TO～，NEXTは上のようにSTEP～を付けると、増加する値を変えることができます。入力し、確かめてください。

いろいろな条件を判断するには

ＹＥＳかＮＯか。この２つの組み合わせでいろいろな条件を判断できる

```
10  CLS
20  INPUT "A=", A
30  INPUT "B=", B
40  C=A/B
50  D=(A MOD B)*1000/B
60  PRINT "A/B=";C;".";D
70  END
```

●答えが小数になる除算を計算する方法

まず、上のプログラムを入力し、実行してみてください。するといつものようにＡとＢの値をたずねてきますから、Ａに３，Ｂに７を入力してみてください。画面は下のようになります。

```
A=3
B=7
A/B=0. 428
OK
■
```

このＢＡＳＩＣでは小数点以下の計算ができないはずなのに、ちゃんと小数点以下３桁まで計算されていますね。

これは、除算の余りを求めるＭＯＤを使っているためです。ＭＯＤは四則計算の記号と同じように、10　ＭＯＤ　４の形で使います。ちなみにこの10　ＭＯＤ　４の計算結果は２になります。

さて、このプログラムでは、50行で小数点以下の計算をしています。このＢＡＳＩＣには小数という数はありませんので、ＡをＢで割ったときの余りを1000倍し、Ｂで割ります。これが小数点以下３桁の値になるわけです。

これで画面に表示させるときに、60行のようにＡをＢで割った答えと、この小数点以下３桁の値を間に「.」を入れて、続けて表示させればいかにもそれらしく見えるわけです。

```
3÷7＝0　余り4
3×1000÷7＝428　余り4
⇨2つを組み合わせて答えを0.428とする
```

●IF～THEN～による条件の判断

63ページのプログラムは除算を小数点以下3桁まで計算してくれるのはいいのですが、50行でAをBで割った余りを1000倍しているため、余りが33以上になったとき、へんな数が表示されるか、エラーとなってしまいます。さきほどのプログラムを消さないで、次のプログラムを入力してみてください。

```
41 IF A MOD B>3276 THEN 60
42 IF A MOD B>327 THEN E=10:GOTO 50
43 IF A MOD B>32 THEN E=100:GOTO 50
44 E=1000
50 D=(A MOD B)*E/B
```

さて、プログラムを実行させて、Aに123，Bに145を入力してみてください。AをBで割った答えは0.84になりました。123を145で割った答えを整数内で考えると0になりますから、余りは123で32を越えています。

しかし、エラーにならないで小数点以下2桁までが計算されました。これはIF～THEN～で、余りの値によって計算する桁数を変えているためです。

IF～THEN～は、IFとTHENの間にある条件が満たされたときにTHEN以降を実行しなさい、という命令です。条件が満たされない場合は、次の行を実行します。

たとえば41行の場合、AをBで割ったときの余りが3276より大きかったら60行を実行し、それ以外の場合は何も行なわれません。

42行や43行も条件を満たした場合のみ、THEN以降を実行する、という点では同じですが、THEN以降がちょっと違いま

す。42行でいえば、E＝10とGOTO　50の２つの命令があり、間を「：」で区切っていますね。

こうすると、１つの行の中で２つ以上の命令を実行させることができるのです。もちろん、THEN以降でなくても使うことができます。また、50行を実行させるのにGOTO　50としていますが、41行のように行番号のみで実行されるのはTHENの直後だけですので、注意してください。

さて、IF～THEN～の役割はわかったと思いますので、かんじんのプログラムの意味を考えてみましょう。

このプログラムは、エラーを出さないA割るBの余りの値の範囲を次のように考えています。

ＢＡＳＩＣが計算で求められる最大値・・・・３２７６７

32767÷1000＝32余り767→小数点以下３桁まで求められるのは０～32まで
32767÷100　＝327余り67→小数点以下２桁まで求められるのは33～327まで
32767÷10　＝3276余り７→小数点以下１桁まで求められるのは327～3276まで

ちょっと考えてみればわかると思いますが、後でAをBで割ったときの余りにかける値と、ＢＡＳＩＣの計算で求められる最大値から、逆算してエラーの出ない範囲を出しているのです。

余りが32767を越える場合も考えられますが、余りが32767になるということは、Aの値は32767以上になりますので、Aの値を入力する時点ですでにアウトです。考える必要はありませんね。

さて、もう一度74ページのプログラムを見てください。今の条件をプログラムにするとこのようになるわけです。

このプログラムは今までのプログラムよりも、少々難しかったかもしれません。しかし、プログラムでＢＡＳＩＣの機能以上のことができる、ということがわかっていただけたと思います。

自分でプログラムを作っているとき、ついついＢＡＳＩＣの機能の範囲内でものごとを考えてしまいがちです。不可能を可能にしてやるんだ、というくらいの意気込みでいけば、自然とプログラムを作る腕も上達するし、いいプログラムができるのです。

カセットの接触が悪くなってしまったら

ファミリーコンピュータのソフトウェアはBASIC用に売られているものをのぞけば全部カセットになっています。したがって、ソフトを変えるときは必ずカセットをぬきさしすることになるわけです。

ていねいにあつかっていればこわれることはありませんが、「はやく次のゲームがやりたい」なんて気持ちがあると、ついつい乱暴になってしまい、エジェクトを思い切りやってしまいがちです。これを繰り返していると、カセットの接触が悪くなってしまい、カセットをさし込んだのに正しく画面が出てこなくなったりすることがあります。

カセットを前後に動かして電源を入れなおすとなおるときがありますが、ますます悪くなることも考えられますので、ひんぱんにやることは進められません。

いちばんいいのは、カセット側の接点をアルコールやCRC5－56などをつけた綿棒で数回こすり、よごれやサビなどを落とします。これでかなり調子がよくなるはずですが、それでもダメなときは販売店や任天堂に相談してみましょう。

なお、ファミリーコンピュータのカセットさし込み口にCRC5－56をふきつけると、カセットのぬきさしが楽になりますが、ホコリが付きやすくなりますので、あまりすすめられません。



FAMILY COMPUTER
FAMILY BASIC
PERFECT BOOK

PART 4

●

ファミリーベーシック
実戦プログラム

今、背景はゲームの大切な要素

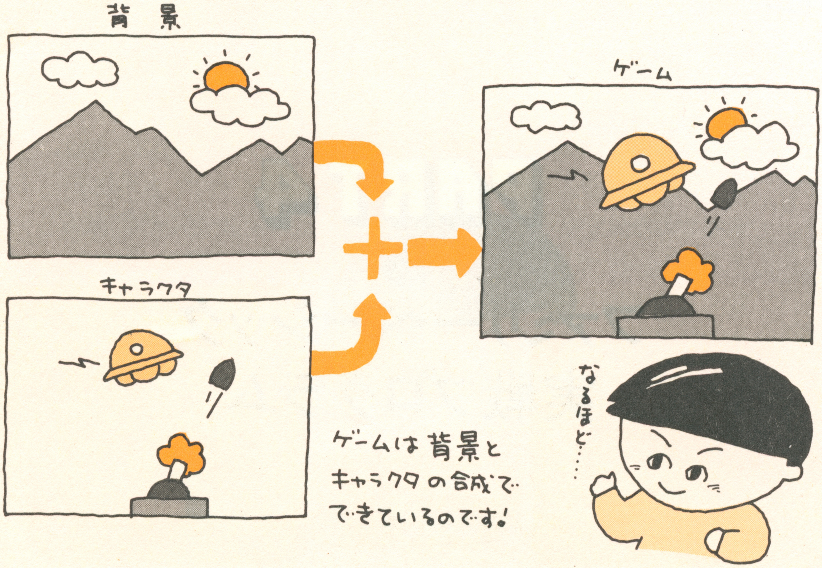
今までのプログラムは、基本的なものでしたので、あまりファミリーベーシックの特徴を生かしたものではありませんでした。

そこで、これからファミリーベーシックの特徴を生かしたプログラムを作ってみましょう。

パソコンのゲームには必ずといっていいほど背景が付いていますね。この背景には大きく分けて２つの種類があると考えられます。

１つはゲームの内容と全くかかわりを持たない、つまり空や山や海など、ゲームを盛り上げ、雰囲気を出すために付けるものです。これは「ないよりはあった方がいい」的な考えから、ゲームのプログラムが完成した後で作る場合が多いようです。

もう１つは、ゲームの内容に直接かかわってくるもの、つまり迷路ゲームの迷路や、レーシングゲームのコースなどです。この場合、背景がなくてはゲームになりませんので、かなり重要な役割をはたしているといえます。特に迷路ゲームの場合、背景となる迷路がゲームのおもしろさそのものを決めてしまいますので、「何でもいいから作ればいい」というわけにはいかないでしょう。



BG GRAPHICで背景作り

ビージー　グラフィック

つまりBack Graund(背景) GRAPHICということである

それでは、何はともあれ背景を作ってみましょう。「プログラムを作らないうちから背景を作っても……」と疑問に思うかもしれませんが、どちらから作ってもかまわないわけですし、簡単なようでけっこう難しいので練習のつもりでチャレンジしてみてください。

さて、背景を作るには、BASICのＰＲＩＮＴを使って作る方法と、ファミリーベーシックに内蔵されているBG GRAPHICを使って作る方法があります。

BASICのＰＲＩＮＴをうまく使って背景を作ると、画面全体の感じがうまくつかめないばかりか、キャラクタの選択や色のつけ方がかなりめんどうです。特に意図するところがないのでしたらBG GRAPHICを使った方が賢明です。

●ＢＧ　ＧＲＡＰＨＩＣ画面にするには？

さて、それではまずBG GRAPHIC画面を出します。BASICを使っている場合は、ＳＹＳＴＥＭと入力し、RETURN キーを押してください。それ以外の場合は、メニュー画面でGAME BASICを選びます。

すると画面はGAME BASICモード画面に変わりますので、2 のキーを押してBG GRAPHICを選んでください。

これで下にあるような、画面が表示されます。これがBG GRAPHIC画面です。

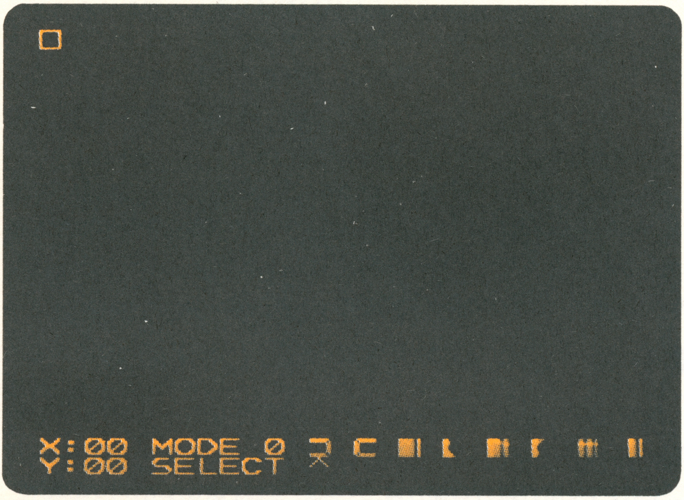
さらに、BG GRAPHIC画面からGAME BASICモード画面にもどりたいときには、ESC キーを押してから STOP キーを押します。

GAME BASICモード画面　　BG GRAPHIC画面



BG GRAPHICの使い方

さて、次のようなBG GRAPHICの画面を表示させることができましたか。



画面の左上に表示されている四角いワクがBG GRAPHICのカーソルです。そして、画面の左下にX：00、Y：00と書かれているのが、それぞれカーソルの横と縦の位置を表示しています。

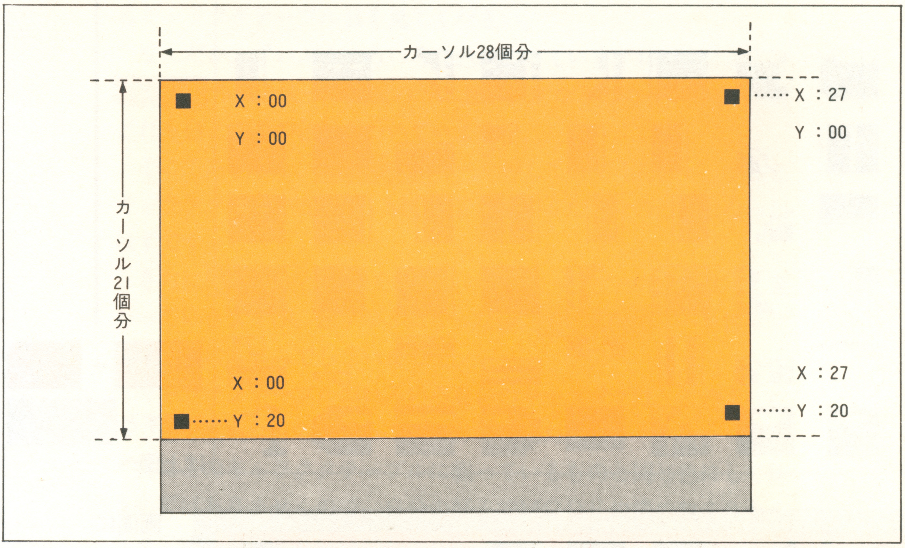
カーソルを動かすには、BASICのときと同じ←→↑↓の4つのキーを使います。ためしに→キーを押してみてください。カーソルが右に1つ動き、X：01がY：00に変わったはずです。

さらに→キーを押していくと、カーソルはどんどん右へ行き、Xの数値もどんどん増えていきます。そして、X：27になるとX：00にもどり、カーソルは再び左側に移ります。

さて、今度は↓キーを押してみましょう。Yの値がだんだん増えていき、Y：20で止まりました。

このことからわかるように、カーソルの動く範囲は横が0から27のカーソル28個分、縦が0から20のカーソル21個分であることがわかります。

● BG GRAPHIC でカーソルの動く位置



さて、それではカーソルを最初のX：00、Y：00の位置にもどしてから、おもむろに スペース キーを押してみましょう。カーソルの位置に、画面下で「へ」マークが点滅している上にあるキャラクタと同じものが表示されたはずです。

さらに スペース キーを押していくと、カーソルが動いて、さきほどと同じキャラクタがどんどん表示されました。

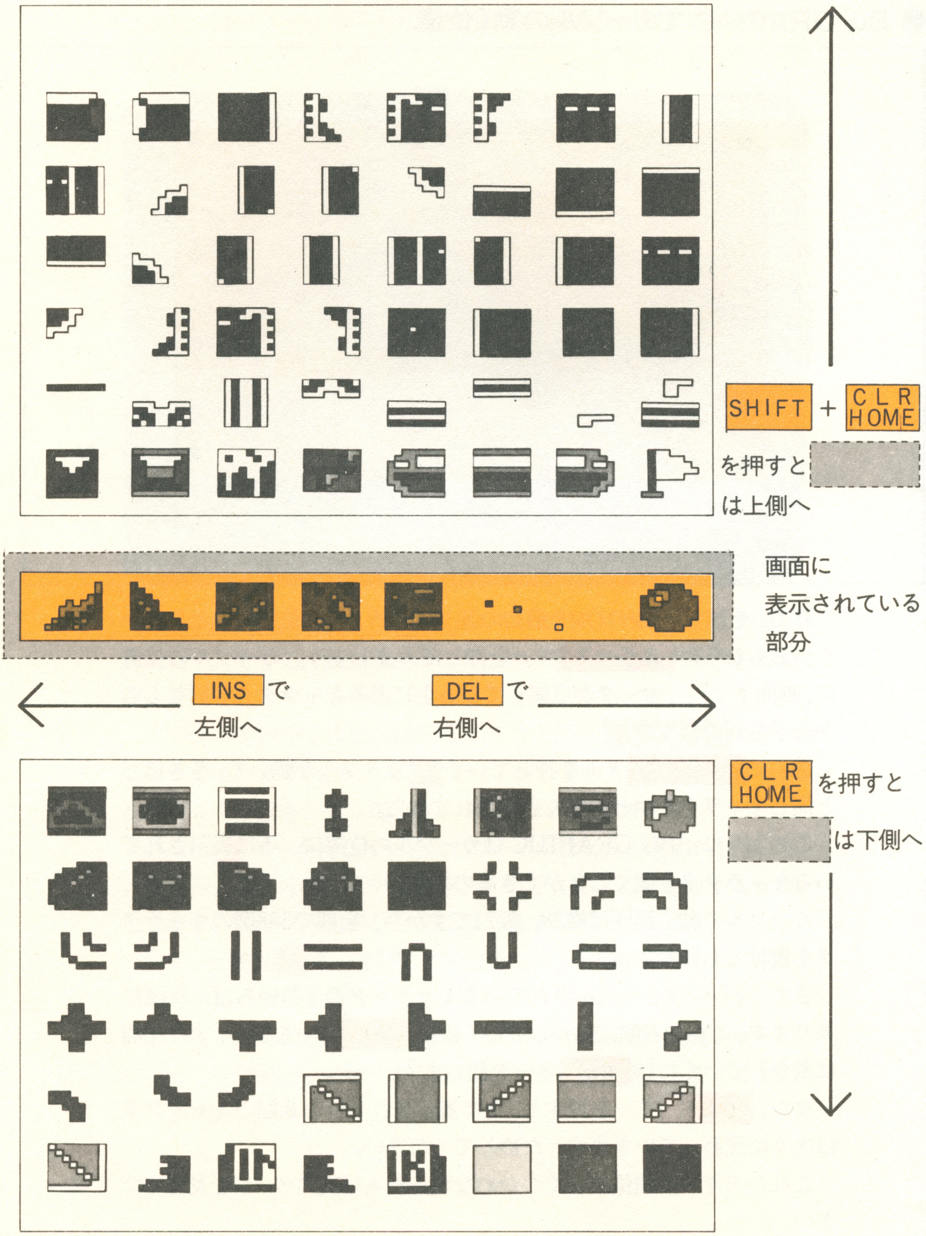
このように、BG GRAPHIC はカーソルの位置に、下に表示されているキャラクタを置くことができるのです。

カーソルの動く範囲は横28、縦21ですから、全部で588個のキャラクタを置けるわけですね。

さて、今「へ」は8つ表示されているキャラクタのうちいちばん左側にあります。これを右側に動かしていくには DEL キーを使い、また左側に動かしていくには INS キーを使います。

また、 CLR キーを押すと、下に表示されている8個のキャラクタは次々に変わっていきます。ためしてください。

これらのキーの関係がよくつかめなかった人は次のページを見てください。


SHIFT + CLR HOME
を押すと
は上側へ
画面に
表示されている
部分
INS で
左側へ
DEL で
右側へ
CLR HOME を押すと
は下側へ

キャラクタの色を変えてみよう

ここでちょっとマニュアルの113ページや、キャラクタマップにあるキャラクタテーブルBを見てください。

なんだか、ずいぶんいろいろな色がついていますね。ところが、HOME キーを使って画面の下に表示させた8つのキャラクタは、どれも白と青系の色しかついていません。

そこでちょっと、RETURN キーを押してみてください。するとキャラクタの色が変わり、MODEの横が0だったのが1になったはずです。

さらに RETURN キーを押していくとキャラクタの色は次々と変わっていきますが、数字が3になった後、RETURN キーを押すと再びもとにもどってしまいました。

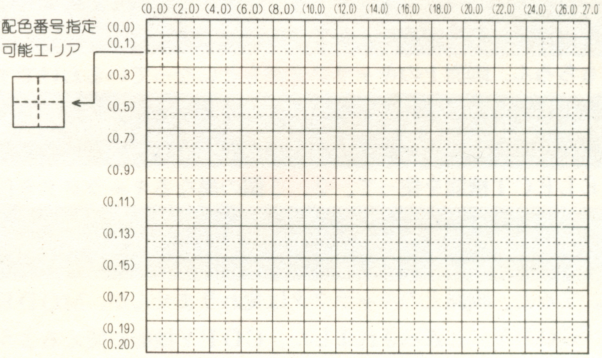
キャラクタテーブルのいろいろな色のキャラクタは、このようにして色を変えたものが載っていたわけですね。

特にこのキャラクタテーブルに載っているものと同じ色にして使う必要はありませんので、自分の好みでMODEの横の数を0から3まで使いわけるといいでしょう。

また、D キーを押すと、カーソルのある場所のキャラクタを消すことができます。

色が変えられる範囲

MODEの横の数字を RETURN キーを押して変えると、キャラクタの色は変わりますが、スペース キーを使って画面に表示させると、下の実線内は同じ色になります。注意してください。



ファンクション機能を表示させよう

BG GRAPHICにはこの他、いろいろな機能があります。ためしに ESC キーを押してみてください。画面の左上に、次のような表示が出ます。

```
>SELECT
 COPY
 MOVE
 CLEAR
 FILE
 CHAR
```

これは、ファンクションメニューと呼ばれるもので、↑ ↓ のキーを使って「>」を動かして、スペース キーで選ぶことにより、それぞれの機能が使えます。

ＦＩＬＥ以外の機能は、すべて画面に関するもので、どの機能を使っているのかは、MODEの下に表示されます。

そういえば今までMODEの下には「ＳＥＬＥＣＴ」と表示されていましたね。これまで使っていたのは、ＳＥＬＥＣＴ機能だったわけです。

また、このファンクションメニューが表示されているときに STOP キーを押すと、GAME BASICモード画面に戻ることができます。

COPY機能

それでは、まずＣＯＰＹ機能から使ってみましょう。

ＣＯＰＹ機能は画面上にあるキャラクタを、同じ色のまま画面上の他の位置に複写することができる機能です。

もちろん、画面上のキャラクタがもとになっているわけですから画面上に何も表示されていないときは使えません。

現在、画面上にキャラクタが１つも表示されていないのでしたら、ＳＥＬＥＣＴ機能を使って、２、３個の異なるキャラクタを色を変えて表示させてからはじめてください。

さて、まずファンクションメニューの中から、ＣＯＰＹを選びます。すると、画面上のキャラクタは変わりませんが、ＭＯＤＥの下に「ＣＯＰＹ」と表示されました。ＣＯＰＹ機能が使えるようにな

ったわけです。
　次に、←→↑↓キーを押して、複写したいキャラクタのところへカーソルを持っていきます。カーソルがキャラクタと重なったら INS キーを押してください。
　そして再び←→↑↓キーを使ってカーソルを動かすと、表示されていたキャラクタはそのまま残り、カーソルのある位置には、それと同じキャラクタが表示されています。
　ここで、DEL キーを押すと、カーソルの位置にそのキャラクタをそのまま置くことができます。そして再び INS キーを押すまで、何回も DEL キーを押して同じキャラクタの複写ができます。
　複写したキャラクタはもとのキャラクタと同じ色になりますが、場所によっては周りのキャラクタの色が変わってしまいます。73ページの「MODEで色が変えられる範囲」をもう一度見て下さい。

①←→↑↓キーでカーソルを複写したいキャラクタの位置までもってくる

↓

② INS キーを押す

↓

③←→↑↓キーでカーソルを複写したい位置までもってくる

↓

④ DEL キーを押す

↓

⑤カーソルの位置にもとのキャラクタと同じキャラクタが置かれる

↓

⑥複写したいキャラクタを換えたいときは①から、同じものの場合は③から行なう

MOVE機能

MOVE機能は、画面上に表示されているキャラクタを他の位置に移動させることができます。

もちろん、画面上に何も表示されていなければ移動させようにも移動させるものがないわけですから、何もできません。SELECT機能を使って適当なキャラクタを表示させた後、行なってください。

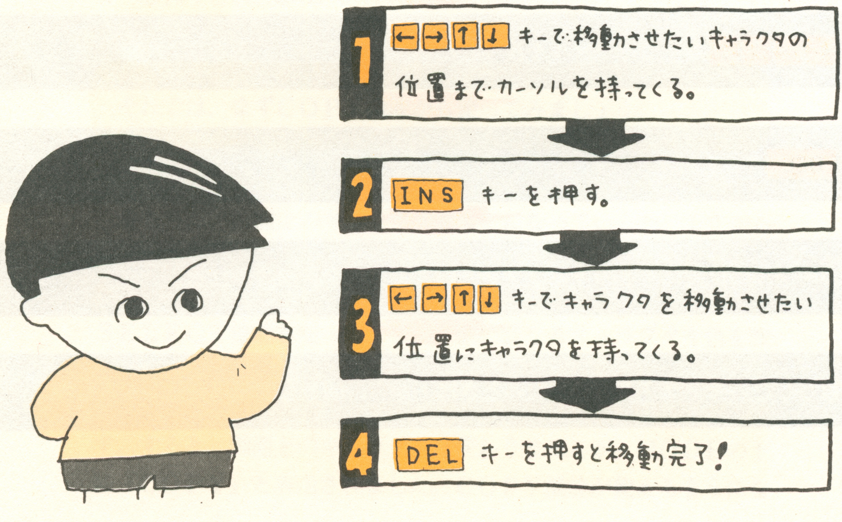
それでは、まずファンクションメニューでMOVEを選びます。MODEの下には「MOVE」と表示されました。

まず、←→↑↓キーを使ってカーソルを移動したいキャラクタの位置まで動かします。カーソルがキャラクタと重なったらINSキーを押してください。

そして、←→↑↓キーでカーソルをキャラクタの移動させたい位置まで持ってきます。このとき、すでにもとの位置にキャラクタはありません。

DELキーを押せば、カーソルの位置にキャラクタが置かれます。

使い方は、ほとんどCOPY機能と同じですが、もとのキャラクタが残らないこと、DELキーだけを何回も押して使えないことが違います。



CLEAR機能

ＣＬＥＡＲ機能は、今までいろいろな方法で画面に表示させたキャラクタを全て消すものです。

ファンクションメニューでＣＬＥＡＲを選ぶと、その時点で今まで表示されていたキャラクタは1つ残らず消えてしまいます。表示されている画面がいらないとき以外は使わないでください。

なおＣＬＥＡＲ機能が使われると、画面が消えた後、自動的にＳＥＬＥＣＴ機能を選びます。

CHAR機能

ＦＩＬＥ機能は、直接画面の表示とは関係ありませんので、先にＣＨＡＲ機能をみてみましょう。

ＣＨＡＲ機能は、画面にキーボードから入力した文字を表示させるためのものです。

ファンクションメニューでＣＨＡＲを選ぶと、ＭＯＤＥの下に「ＣＨＡＲ」と表示され、カーソルが□から BASIC と同じ■に変わります。

これで、画面にキーボードから入力した文字が表示できるようになったわけです。ためしにＰＲＩＮＴ２＊３と入力し、RETURNキーを押してみてください。

```
PRINT    2*3
■
```

おや？　カーソルは次の行の先頭に移動しましたが、２＊３の計算結果は表示されませんね。

ＣＨＡＲ機能は画面にキーボードから入力された文字を表示するだけですから、それが BASIC の命令であろうがなかろうが全く関係ないのです。気にせず好きなことを入力してみてください。

なお、ＣＨＡＲ機能では下の５つのキーは使えません。

ＣＨＡＲ機能で使えない５つのキー：
HOME　INS　DEL　STOP　CTR

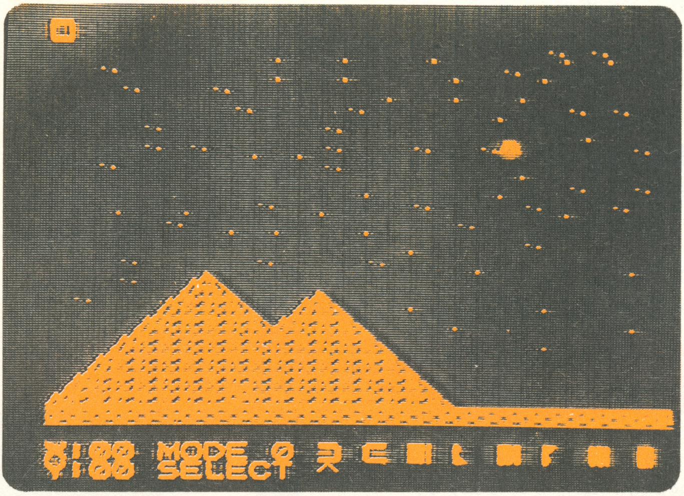
FILE機能

ＦＩＬＥ機能は、BG GRAPHIC で作った画面をカセットテープに記録したり、すでにカセットテープに記録されている画面を読み込むためのものです。

BG GRAPHIC で作った画面は、バックアップスイッチを使って保存することはできません。バックアップスイッチをＯＮにしておいても作った画面は消えてしまいます。

ですから、BG GRAPHIC で作った画面を保存するには、カセットテープに記録するしか方法がないのです。

とりあえず、画面に何も表示されていなくては、記録してもしょうがないので、次のような画面を作ってみました。



この画面は使っているキャラクタが限られていますので、最初にＳＥＬＥＣＴ機能で使用するキャラクタを色を決めて、１つずつ表示させておきます。そして、ＣＯＰＹ機能を使えば簡単に作れてしまいます。

さて、それでは ESC キーを押して、ファンクションメニューを表示させ ↓ キーで「>」をＦＩＬＥのところに移動させてから スペース キーを押してください。画面には次のように表示されます。

```
SAVE (S), LOAD (L)?■
```

ここでは、テープに記録するのでSを押します。画面には次のようにファイルネームをたずねてきます。

```
SAVE (S), LOAD (L)?S
FILE  NAME "■
```

BACK1と入力し、データレコーダにカセットテープをセットして録音ボタンを押してから RETURN キーを押します。

```
SAVE (S), LOAD (L)?S
FILE  NAME "BACK1
WRITING  BACK1
```

カセットテープに記録しているときは上のように表示されますが、終了すると自動的にSELECT機能に移ります。しかし、さきほど作った画面はそのまま残っています。FILE機能には、BASICのLOAD?のような正しく記録されているかどうかを確かめるような便利なものはついていません。不安になるようでしたら、もう1回くらいテープに記録しておいてください。カセットテープに記録しておいた画面を読み込むには、FILE機能を選んで、「SAVE(S), LOAD(L)?」が表示されているときLを押します。記録のときと同じく、ファイルネームを聞いてきますので、さきほど入れたBACK1を入力し、 RETURN キーを押します。

正しく読み込みができれば、次のように4回表示された後、SELECT機能が選択されます。

```
LOADING  BACK1
```

うまく読み込めなかったら、40ページを参考にしてください。

使えるアニメ・キャラクタを全部表示してみよう

この BASIC の最大の特徴は、画面に表示したアニメキャラクタが簡単に動かせることです。このアニメキャラクタは、スプライトと呼ばれている、キーボードから入力した文字と違う画面に表示されます。

前にマリオが動くプログラムがありましたが、これはスプライトを利用して動かしたものだったのです。

それでは、このスプライトを使って、アニメキャラクタの動かし方を見てみましょう。いつものように、プログラムを入力し、実行してから説明を読んでください。

●アニメ・キャラクタ全表示プログラム

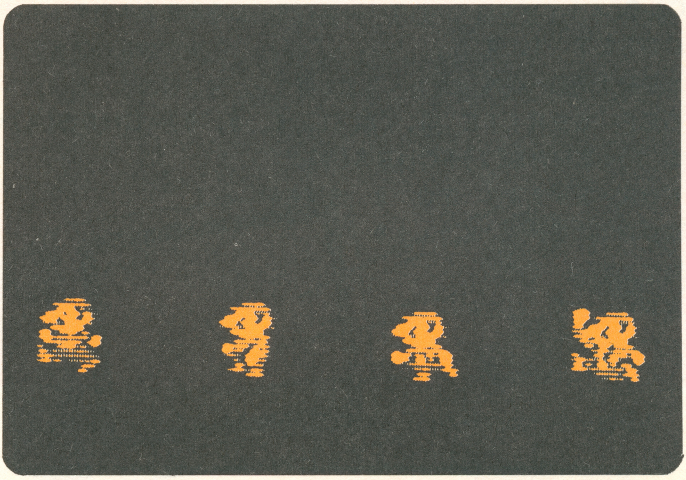
```
10 CLS
20 CGSET 1,0
30 SPRITE ON
40 FOR S=0 TO 207 STEP 16
50 FOR C=0 TO 15 STEP 4
60 DEF SPRITE C/4,(0,1,0,0,0)=CHR$(S+
C)+CHR$(S+C+1)+CHR$(S+C+2)+CHR$(S+C+3)
70 SPRITE C/4,C*8+50,100
80 NEXT
90 PAUSE
100 NEXT
110 SPRITE OFF
120 END
```

ちょっと長いですが、上のプログラムを間違わないように入力してから、RUNと入力し、RETURN キーを押してプログラムを実行させてみましょう。

次のページのように、画面上にアニメキャラクタが4つ表示された上にカーソルが出ましたか。

もし、このような画面にならなかったら、プログラムの入力間違

いです。 正しく直してください。



ちょっとマニュアルの裏表紙や、キャラクタマップにあるキャラクタテーブルAを見てください。

今、画面に表示されているアニメキャラクタは、いちばん上の段の左から4つですね。

それでは、次にキーを1回押してみてください。今度は、画面に次の4つが表示されました。

さらにキーを押していくと、最後のカニさんが2つ、トリさんが2つ表示されるまで、4つずつアニメキャラクタを表示していきます。そしてプログラムは終わり、画面上にはOKが表示されました。

今、画面上に表示されたのが、このBASICで使えるアニメキャラクタの、ほぼ全部です。画面に表示されたものの他にも、レーザーや、キャラクタテーブルの表に載っていないものがありますが、それについては、もう少し後でふれることにしましょう。

また、今画面に表示されたアニメキャラクタは、キャラクタテーブルに載っているものと、色が違うものがあります。その理由は、後にある「アニメキャラクタの色を変えてみよう」というところを読めばわかります。

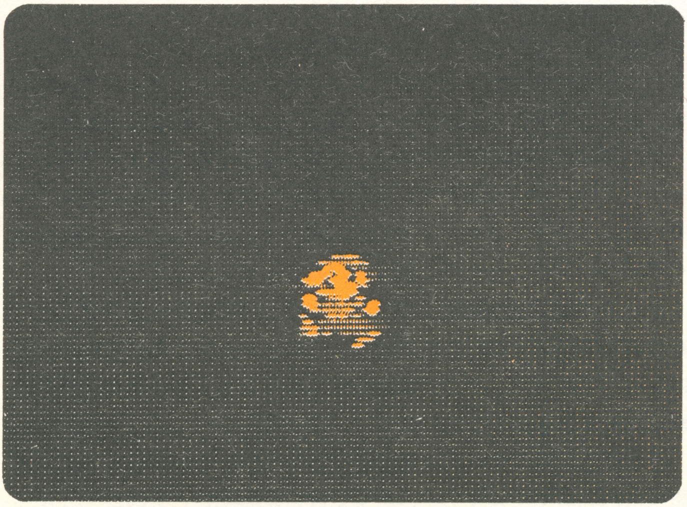
どうしても知りたい人は先に読んでください。

アニメ・キャラクタを動かそう Part1

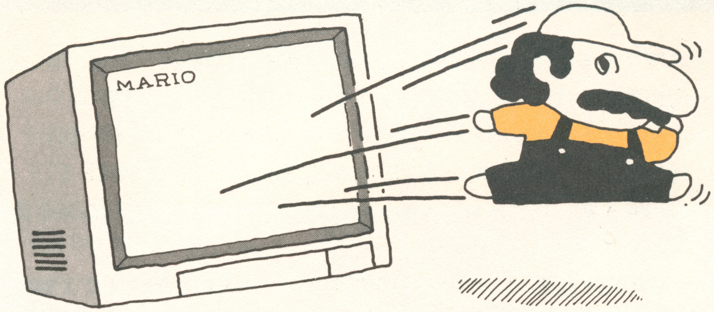
```
10 CLS
20 CGSET 1, 0
30 SPRITE ON
40 DEF SPRITE 0,(0,1,0,0,0)=CHR$(0)+
CHR$(1)+CHR$(2)+CHR$(3)
50 FOR X=240 TO 0 STEP -1
60 SPRITE 0,X,100
70 NEXT
80 SPRITE OFF
90 END
```

さきほどのプログラムは、必要でしたらカセットテープに記録しておき、とりあえずNEWでメモリ上から消してから上のプログラムを入力します。間違いなく入力できましたら、RUNでプログラムを実行させてみましょう。

次のように，マリオが画面の右から左へ向かってものすごいスピードで動いていったはずです。



これは、DEF　SPRITEとSPRITEを使って、マリオのキャラクタを表示させ、FOR～　TO～，NEXTでその位置を変えているのです。
それでは、このDEF　SPRITEとSPRITEの使い方をみてみましょう。



●DEF SPRITEは アニメ・キャラクタを定義する命令だ

まず、40行を見てください。ここにあるDEF　SPRITEというのが、スプライト面に表示させるスプライトにキャラクタテーブルAにあるアニメキャラクタを定義する命令なのです。

スプライト面は、キーボードから入力した文字が表示される画面とは全く別にある画面です。

このスプライト面には、文字などを含むキャラクタを表示させるためのスプライトというものがあります。

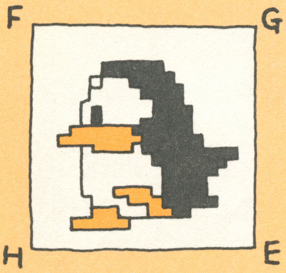
はじめ、このスプライトは透明になっています。ですから、DEF　SPRITEを使ってアニメキャラクタなどを定義するわけです。

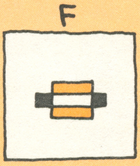
40行をみてもわかるように、DEF　SPRITEの後ろには、何やらいろいろなものがたくさんくっつきます。これらは次にあるようなことを意味しています。

●DEF　SPRITEの使い方

```
DEF　SPRITE n, (A, B, C, D, E)
=CHR$(F)+CHR$(G)+CHR$(H)+CHR$(I)
```

n……スプライト番号。0～7の8つのスプライトに対応
A……配色番号。カラーチャートにある0～3の配色番号に対応
B……キャラクタの合成型。キャラクタテーブルAにあるレーザーは0、それ以外は1。これが違っていると、うまく表示されないことがある
C……表示優先度。0だとスプライト面はキーボードから入力した文字より手前、1だと後側に表示される
D……X軸方向反転表示。0でキャラクタテーブルと同じ形、1でX軸方向に対し反転したキャラクタになる
E……Y軸方向反転表示。0でキャラクタテーブルと同じ形、1でX軸方向に対し反転したキャラクタになる
F……
G……　キャラクタテーブルAにあるキャラクタを囲んだ正方形の
H……　かどにある数字に下のように対応させる
I……





＊この場合、残りの3つのCHR＄は不要で＝CHR＄(F) とすればいい

DEF　SPRITEでアニメキャラクタを定義する方法のなかで、X軸方向反転表示とY軸反転表示というのがありましたが、これは頭で考えただけではちょっとわからないかもしれません。

マニュアルを読んで理解できた人はともかく、よくわからなかった人は次のプログラムを入力してください。

```
10 CLS
20 CGSET 1,0:SPRITE ON
30 LOCATE 0,0
40 INPUT "D=",D
50 IF D<0 OR D>1 THEN 30
60 LOCATE 0,1
70 INPUT "E=",E
80 IF E<0 OR E>1 THEN 60
90 DEF SPRITE 0,(0,1,0,D,E)=CHR$(0)+CHR$(1)+CHR$(2)+CHR$(3)
100 SPRITE 0,128,100
110 INPUT "ツヅケマスカ (Y OR N)";C$
120 IF C$="N" THEN SPRITE OFF:END
130 GOTO 30
```

このプログラムを実行させると、DとEの値をたずねてきます。このDとEというのは、84ページの「DEF　SPRITEの使い方」の中にあるものと全く同じ役目をはたしています。ですから、それぞれに0か1を入力してみてください。入力が終わると画面中央にその場合について定義されたキャラクタが表示されます。

表示が終わると続けるかどうかを聞いてきますので、Yを入力していろいろな場合について調べてみましょう。

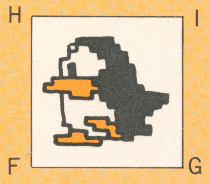
さて、調べた結果はどうでしたか。D＝0、E＝0以外の場合はちゃんとした形になったはずですが、それ以外の場合は、形のくずれたわけのわからないキャラクタが表示されましたね。

これでわかったように、このDとEの値を変えただけでは、アニメキャラクタを横向きにしたり、下向きにしたりすることはできま

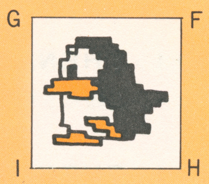
せん。それには、もうひとくふう必要なのです。次の図を参考にしてください。

●反転キャラクタを定義するには

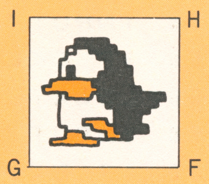
上下反転
D＝0
E＝I



左右反転
D＝I
E＝0



上下左右反転
D＝I
E＝I



(例) マリオの場合だと次のようになる

●上下方向反転

```
DEF SPRITE 0,(0,1,0,0,1)=CHR$(2)+
CHR$(3)+CHR$(0)+CHR$(1)
```

●左右方向反転

```
DEF SPRITE 0,(0,1,0,1,0)=CHR$(1)+
CHR$(0)+CHR$(3)+CHR$(2)
```

●上下左右方向反転

```
DEF SPRITE 0,(0,1,0,1,1)=CHR$(3)+
CHR$(2)+CHR$(1)+CHR$(0)
```



●SPRITEはアニメ・キャラクタを表示させる命令だ

DEF　SPRITEはスプライトにアニメキャラクタを定義する命令でしたが、定義しただけでは画面にアニメキャラクタは表示されません。

画面にアニメキャラクタを表示させるには、SPRITEという命令を使います。

DEF　SPRITEを使って定義できるアニメキャラクタは8個でしたがSPRITEを使って1度に画面上横1列に表示できるのは、そのうちの4つまでです。80ページの「使えるアニメキャラクタを全部表示してみよう」で、アニメキャラクタが画面上に4つずつしか表示されなかったのは、そのためです。

その他の場合は8つ表示できます。

さて、SPRITEは次のように使います。

●SPRITEの使い方

SPRITE　n,　X,Y

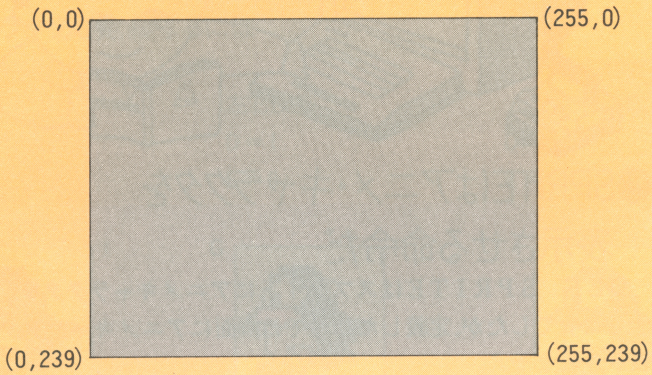
n……スプライト番号。DEF　SPRITEで定義したときのスプライト番号に対応

X……スプライトを表示する横方向の座標。0〜255

Y……スプライトを表示する縦方向の座標。0〜255

X,Yは0〜255の値であればエラーにならないが、実際の画面とは次のような関係がある

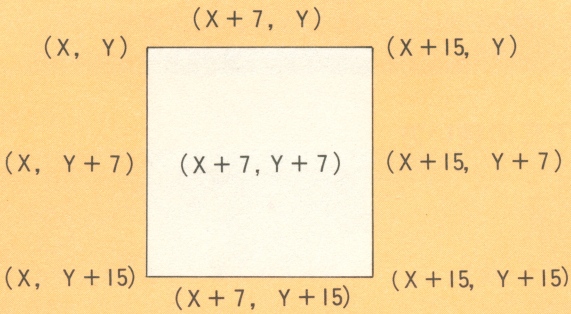
スプライト面の座標（X，Y）



SPRITEとアニメキャラクタの関係

ＳＰＲＩＴＥ　n，X，Yとした場合、アニメキャラクタの場所によって座標の値は異なる

●ＳＰＲＩＴＥ　n，X，Yに対するアニメキャラクタの座標（x，y）



したがってＳＰＲＩＴＥ　n，X，Yでの X，Yの有効となる範囲は

X……0～２４０

Y……0～２２４

{注：テレビによって多少異なることがある}

アニメ・キャラクタを動かそうPart2

```
10 CLS
20 CGSET 1,0
30 SPRITE ON
40 DEF MOVE(0)=SPRITE(0,7,1,120,0,0)
50 POSITION 0,240,100
60 MOVE 0
70 END
```

アニメキャラクタを動かすには、もう1つの方法があります。まず、上のプログラムを入力し、実行させてみてください。

すると、さきほどのDEF SPRITEとSPRITEを使ったときと同じように、マリオが右から左へ動いていきました。でもさっきとちょっと違いますね。どこが違うかわかりますか。

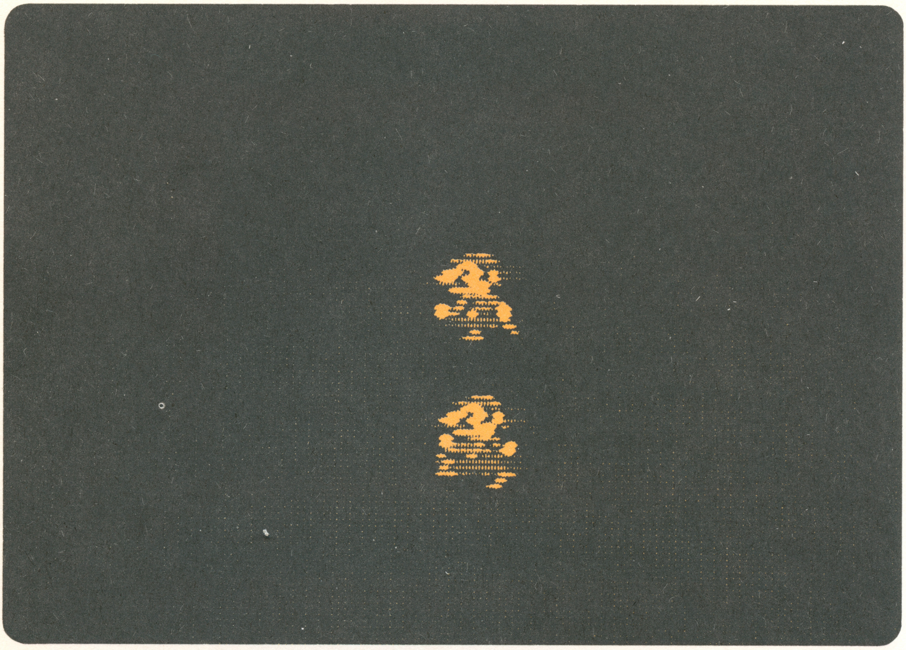
動きが遅い!? まあそれもありますが、それだけではないはずです。

まず1つは、プログラムが終わってOKとカーソルが表示されているのにマリオが動いていること、もう1つはマリオが手足を動かしながら動いていることです。

それでも違いがよくわからない、という人は、上のプログラムを消さないで次のプログラムを入力させてみてください。

```
45 DEF SPRITE 0,(0,1,0,0,0)=CHR$(0)
+CHR$(1)+CHR$(2)+CHR$(3)
70 SPRITE 0,XPOS(0),125
80 IF MOVE(0) THEN 70
90 SPRITE OFF
100 END
```

このプログラムは、さきほど動いていたマリオのちょっと下に、DEF SPRITEとSPRITEを使って、同じ速さでもう1つのマリオを動かすためのものです。入力が終わっていたら実行させてみましょう。画面は次のようになります。



違いがわからなかった人も、今度はよくわかったでしょう。上のマリオは手足を動かしながら動いていきますが、下のマリオは片足を上げて止まったまま動いていますね。

さて、それではOKとカーソルが出てからもう一度プログラムを実行させてみますが、今度はマリオが画面の中央付近に近づいたときに、STOPキーを押してプログラムを止めてみてください。

すると、画面の左上にはBREAK　IN　70と出て、OKとカーソルが表示されました。プログラムが70行で止まった、ということですね。

ところが、画面上のマリオで止まっているのは下の方だけで上の方はまだ動いています。そして、画面の左すみまで動いていき、やっと止まりました。

以上がDEF　MOVEとMOVEを使ってアニメキャラクタを動かしたときの特徴です。それでは、このDEF　MOVEとMOVEを使い方を見てみましょう。

●DEF MOVEはアニメ・キャラクタの動作を決める命令だ

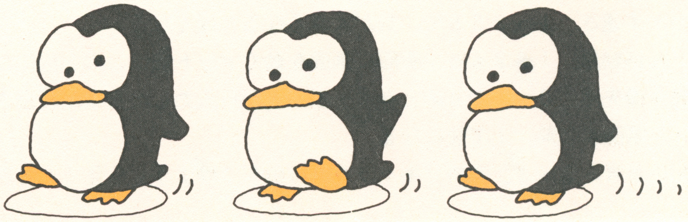
ＤＥＦ　ＭＯＶＥとＭＯＶＥを使ってアニメキャラクタを動かす場合、基本的には、ＤＥＦ　ＳＰＲＩＴＥとＳＰＲＩＴＥを使った場合と同じでスプライトを使っています。したがって、アニメキャラクタはスプライト面に表示されます。

さて、ＤＥＦ　ＳＰＲＩＴＥはアニメキャラクタそのものをスプライトに定義しますが、ＤＥＦ　ＭＯＶＥはアニメキャラクタの種類を選ぶことで数種類のアニメキャラクタがスプライトに自動的に定義されます。その点でこの２つは大きく違うといえます。

また、ＤＥＦ　ＳＰＲＩＴＥは、ＳＰＲＩＴＥを使ってアニメキャラクタを画面に表示させるとき、プログラム的に表示させる座標を変えてやらないとアニメキャラクタを動かすことはできませんでした。

しかし、ＤＥＦ　ＭＯＶＥは動作の方向さえ決めてやれば、ＭＯＶＥを使って簡単にアニメキャラクタを動かすことができます。しかも、数種類のアニメキャラクタを勝手に使ってくれますので、ただ移動していくだけでなく、アニメキャラクタ自体を動いているように見せてくれるのです。動く速さがちょっと遅いことをのぞけば、かなり便利な命令ですね。

それでは、次のＤＥＦ　ＭＯＶＥの使い方を見てみましょう。



●DEF　MOVEの使い方

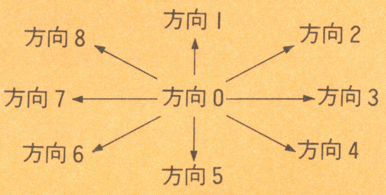
DEF　MOVE (n)=SPRITE (A, B, C, D, E)

n…アニメキャラクタの動作番号。0～7の8つの指定ができる

A……アニメキャラクタの種類。0～15の16種で次の通り

0：マリオ
1：レディ
2：ファイターフライ
3：アキレス
4：ペンペン
5：ファイアーボール
6：車
7：スピナー
8：スターキラー
9：スターシップ
10：爆発
11：ニタニタ
12：レーザー
13：シェルクリーパー
14：サイドステッパー
15：ニットピッカー

B……動作の方向指定。
0～8の9方向で次の通り



C……動かす速さ。1～255で、数が大きくなると遅くなる

D……全体移動量。1～255で実際に画面を動くドット数はこの2倍になる

E……表示優先度。0～1で0のときキーボードから入力した文字が表示される
0で画面より前。1で後になる

F……配色番号。0～3。次の「キャラクタに色をつけよう」を見てください

このDEF　MOVEを使ってキャラクタや、その動きを決めたものを、実際に画面上で動かすにはMOVE文を使います。MOVEの使い方は次の通りです。

●MOVEの使い方

MOVE　n_0〔n_1, n_2, n_3, n_4, n_5, n_6〕

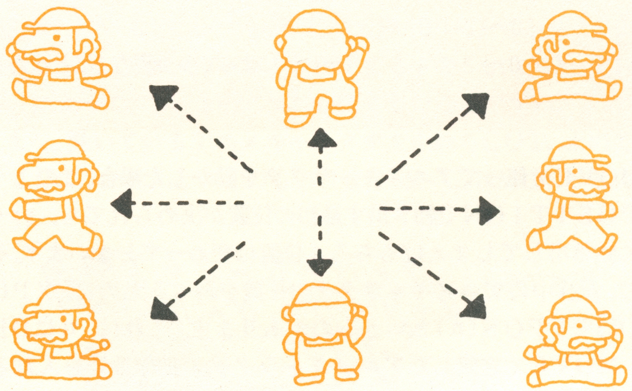
n_0〜n_7…DEF　MOVEであらかじめ定義したアニメキャラクタの動作番号で0〜7

このMOVEは、8種類定義した全部のアニメキャラクタを画面上に表示させることができます。

ただし、画面の横方向に対しては最高4つまでしか表示できず、それ以上表示させようとしても消えてしまいます。注意してください。DEF　MOVEとDEF　SPRITEを使用すれば、合計16個のアニメキャラクタが表示できます。

また、MOVEでは、SPRITEのように画面上のどこに表示させるかということは決められません。画面上のどこから動かすかをきめるのには、POSITIONを使います。

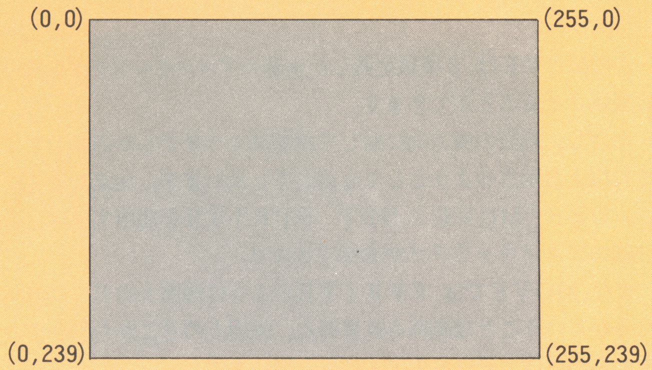
POSITIONで決められる位置はもちろんスプライト面上ですので、次のようになります。



●ＰＯＳＩＴＩＯＮの使い方

ＰＯＳＩＴＩＯＮ　n，X，Y，

n…ＤＥＦ　ＳＰＲＩＴＥであらかじめ定義したアニメキャラクタの動作番号，０～７
X…スプライト面の横方向の座標。０～255
Y…スプライト面の縦方向の座標。０～255
実際のスプライト面と座標（X，Y）は次のように対応している



（X，Y）はＳＰＲＩＴＥのときと同じく、アニメキャラクタの左上の座標になるので、実際、画面上に表示させることができる範囲は次の通り

X…０～２４０
Y…０～２２４　（テレビによって多少異なることがある）

ＭＯＶＥを使ってアニメキャラクタを動かした場合、このようにＰＯＳＩＴＩＯＮで動き出す最初の位置が決められても、いったん動き出してしまえばもちろん位置は変わってしまいます。そこで、動作中のアニメキャラクタの位置を調べるために、ＸＰＯＳ(n)とＹＰＯＳ(n)というものがあります。これは、nにＤＥＦ　ＭＯＶＥで定義したアニメキャラクタの動作番号を入れる

と、そのアニメキャラクタが現在画面上のどの位置にあるかを示してくれます。それでは、次のプログラムを入力してみてください。

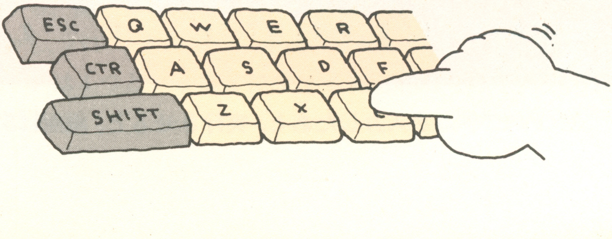
```
10 CLS
20 CGSET 1,0
30 SPRITE ON
40 DEF MOVE(0)=SPRITE(0,7,3,120,0,0)
50 POSITION 0,240,100
60 MOVE 0
70 LOCATE 0,0:PRINT "X=";XPOS(0);" "
80 LOCATE 0,1:PRINT "Y=";YPOS(0);" "
90 IF MOVE(0)=-1THEN 70
100 SPRITE OFF
110 END
```

このプログラムを実行させると、画面の左上に現在動いているアニメキャラクタの表示位置を示してくれます。

さて、90行でMOVE(0)というのをIF～THEN～の判断部分に使っています。これはカッコの中に入れた動作番号のアニメキャラクタが、DEF　MOVEで定義した移動量分だけ動いたかを調べるためのものです。動作が完了していなければ偽の値の0，完了していれば真の値の－1になります。ですから90行のMOVE(0)＝－1をMOVE(0)としてもよいわけですね。

動かしているアニメキャラクタは、動作が完了するまで止まりませんから、アニメキャラクタが止まるまで、このIF～THEN～によって、70、80行の実行が続くわけなんですね。

また、動いているアニメキャラクタを止めるCUT，止めて消してしまうERAというのもあります。使い方はMOVEと同じです。



キャラクタの色を変えてみよう

```
10 CLS
20 SPRITE ON
30 FOR I=0 TO 3
40 DEF MOVE(I)=SPRITE(0, 7, 1, 128, 0, I)
50 POSITION I, 240, I*22+60
60 NEXT
70 FOR C=0 TO 2
80 CGSET 1, C
90 MOVE 0, 1, 2, 3
100 LOCATE 0, 0
110 PRINT "CGSET 1," ;C
120 IF MOVE(0) THEN 120
130 NEXT
140 CLS : SPRITE OFF
150 END
```

今までCGSETとか配色番号といったものを無条件で使ってきました。実は、これが画面に表示するキャラクタの色を決めるものだったのです。

それでは､上のプログラムを入力し､実行させてみてください。すると、画面の左上に「CGSET 1,0」と表示され、4つのマリオが画面の右側から左側へ向かって歩き出しました。4つとも違う色がついていますね。さらに画面左上には「CGSET 1,1」「CGSET 1,2」と表示されていき､いろいろな色のマリオが歩いていきます。 動いているので､どうも色の違いがよくわからない、と思った人は、次のプログラムを入力してみましょう。

```
40 DEF MOVE(0)=SPRITE(0, 7, 1, 1, 0, I)
50 POSITION I, 72+I*32, 100
115 LOCATE 7, 8:PRINT "0     1     2     3"
120 PAUSE
```

このプログラムを入力したあと、実行させると、画面は次のよ

うになります。

さきほどのときと同じように、画面の左上には「CGSET　1,0」の表示が出ましたが、４つのマリオは画面の中央で止まっていて、その上に０～３の数字が表示されました。



そこでキーボードのキーをどれでもいいですから１回押してみてください。画面上の「CGSET」の後ろの数値が変わり、４つのマリオの色も変わりました。

さらにキーを押していけば、画面左上にあるCGSETの後ろの数値が変わり、「CGSET　1,2」となるまで４つのマリオの色は変わり続けます。さきほど、色の変わりぐあいがよくわからなかった人も今度はよくわかったでしょう。

今、画面の表示されていたCGSETの後の数値がパレットコードと呼ばれるもので、マリオの上に表示されている数値が彩色番号と呼ばれているものです。

パレットコードは０から２まで、彩色番号は０から３までありますから、使える色のパターンは全部で12あることになります。

しかし、パレットは１つしか選べませんので、一度に使える色パターンは４つということになります。くわしくは次の説明を見てください。

●CGSETの使い方

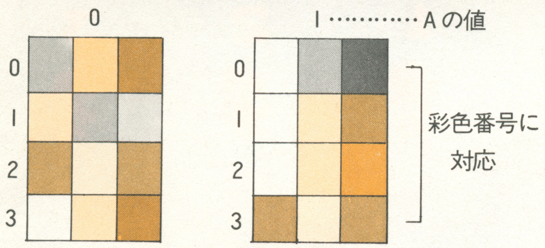
CGSET　A，B

A…キーボードから入力される文字のある面（バックグラウンド面）用のパレットコード。0～1

B…スプライト（アニメキャラクタ）用のパレットコード。0～3

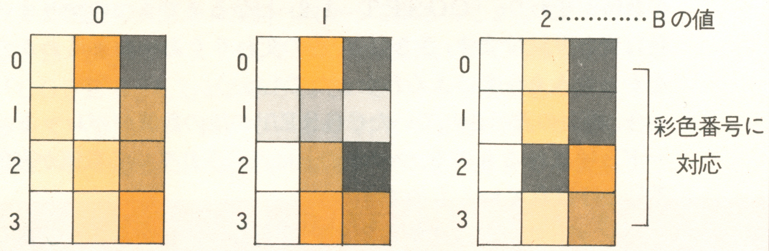
パレットコードと彩色番号は次のような関係になっている

バックグラウンド用



□は1つの色をさすので、1つの彩色番号を選べば、3色使えることになる。したがって、同時に使える色は12パターンということ。通常はパレットコード1が選ばれている

スプライト用



通常はパレットコード1が選ばれている。

＊実際にどんな色が決定されているかは、マニュアルの113ページのキャラクタマップにあるカラーチャートを見てください。

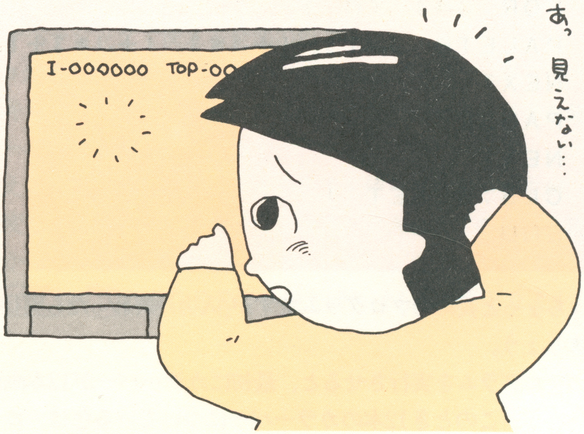
ここで、バックグラウンドということばが出てきました。これはキーボードから入力した文字が表示される画面です。バックグラウンド用のパレットコードを変えても、ＢＧ　ＧＲＡＰＨＩＣで使う色に変化は起こりません。必要がなければパレットコードは１にしてください。

また、各パレットにある色は、最初マニュアルの113ページや、キャラクタマップのカラーチャートにあるように設定されています。カラーチャートにあるアニメキャラクタと同じ色のアニメキャラクタにするには、ＣＧＳＥＴでパレットコードを選んだ後、ＤＥＦ　ＳＰＲＩＴＥやＤＥＦ　ＭＯＶＥでアニメキャラクタのある場所の配色番号を設定します。

この方法で、キャラクタテーブルＡにあるすべてのアニメキャラクタと同じ色のアニメキャラクタが設定できます。

しかし、違うパレットコードにある配色番号を使う２つのアニメキャラクタを、画面上で同時に表示させようとする場合は、今のままでは不可能です。

次のＰＡＬＥＴを使えば可能になりますが、使い方をよく理解してから行なわないと、色がメチャメチャになってしまいます。特にバックグラウンド用の色をＰＡＬＥＴで変えると、文字が見えなくなってしまうことがあります。注意してください。



使える色を全部表示してみよう

ファミリーベーシックは52色の色が使えますが、アニメキャラクタを表示させたりしているだけでは本当に52色も出ているのかよくわかりません。

そこで、ファミリーベーシックで使える52色を全部見てみたい、と思った人は次のプログラムを入力してみましょう。このプログラムは、カラーバーを出すためのものです。

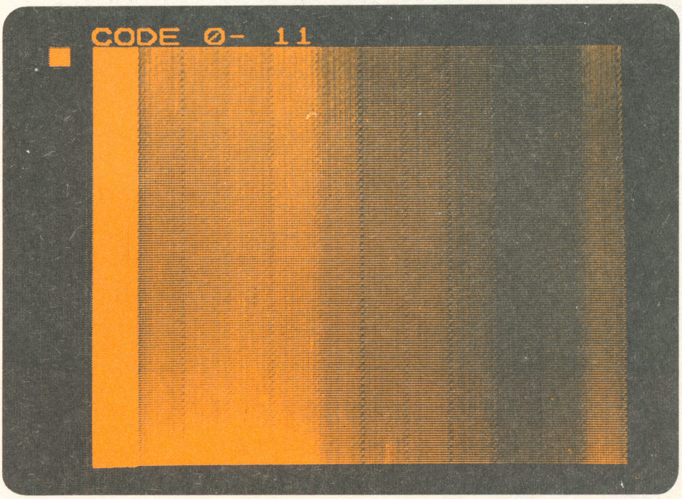
```
10 CLS
20 CGSET 0
30 FOR X=0 TO 18 STEP 6
40 FOR C=0 TO 4 STEP 2
50 FOR Y=1 TO 22
60 COLOR X+C+2, Y, X/6
70 LOCATE X+C+2, Y
80 PRINT CHR$(C/2+253);CHR$(C/2+253)
90 NEXT
100 NEXT
110 NEXT
120 FOR I=0 TO 60 STEP 12
130 LOCATE 2, 0
140 PRINT "CODE"; I ; "-"; I+11
150 FOR J=0 TO 3
160 PALETB J, 15, I+J*3, I+J*3+1, I+J*3+2
170 NEXT
180 PAUSE
190 NEXT
200 CLS:CGSET 1
210 END
```

ちょっと長めのプログラムですが、入力が終わったら実行させてみましょう。

プログラムを実行させると、最初にカラーバーが12本画面に描かれます。このとき12本のカラーバーについている色は、カラチャー

トにあるパレットコード0の色と同じものです。

カラーバーが全部描かれると、パッと色が変わり、次のように画面の上の方に「CODE　0-11」と表示されます。



このとき、キーボードのキーをどれか1つ押すとカラーバーの色が変わり、画面の上に表示されていたCODEも12－23になるはずです。

さらにキーを押していけば、色はつぎつぎに変わり、CODE60－71になるまでカラーバーの色が変わり続けます。今表示されたのがファミリーベーシックで使える全部の色です。

このプログラムはパレットコード0の彩色番号に3色ずつ色を付けていき、キャラクタテーブルBにある最後の3つのキャラクタを使ってカラーバーを表示させています。

色を変えるにはCOLORとPALETを使って行ないますが、かなり大変です。マニュアルのCOLORとPALETの説明を読んでもわからなかったら使わない方がいいでしょう。

なお、CTR＋DかCGSETを使えば色はもとにもどります。

コントローラーを使ってみよう

```
10 CLS:CGSET 1,1:SPRITE ON
20 DEF SPRITE 0,(0,1,0,0,0)=CHR$(172)
  +CHR$(173)+CHR$(174)+CHR$(175)
30 DEF MOVE(0)=SPRITE(12,1,1,100,0,3)
40 DEF MOVE(1)=SPRITE(10,0,4,3,0,3)
50 DEF MOVE(2)=SPRITE(8,7,1,128,0,1)
60 TX=240:TY=30:POSITION 2,TX,TY
70 SX=112:SY=200
80 SPRITE 0,SX,SY
90 IF STICK(0)=2 THEN SX=SX-1
100 IF SX<0 THEN SX=0
110 IF STICK(0)=1 THEN SX=SX+1
120 IF SX>220 THEN SX=220
130 IF MOVE(0) THEN 150
140 IF STRIG(0)=8 THEN POSITION 0,
    SX+4,SY:MOVE 0
150 MOVE 2
160 IF YPOS(0)=28 THEN IF XPOS(0)>XPOS
    (2) AND XPOS(0)<XPOS(2)+16 THEN 180
170 GOTO 80
180 ERA 0,2
190 POSITION 1,XPOS(2),YPOS(2):MOVE 1
200 LOCATE 3,10:INPUT "TRY AGAIN(Y OR
    N)";TR$
210 IF TR$<>"N" THEN RUN
220 CLS:SPRITE OFF:CGSET 1,1
230 END
```

それではいつものように、上のプログラムを入力してください。
かなり長いプログラムですので、間違えないように気をつけましょう。



さて、プログラムを実行させると、画面の下中央にスターシップが表示され、画面上方にスターキラーが右から左へ動いていきます。

そこでためしにコントローラーⅠの✚を右側か左側に押してみてください。すると、下の方に表示されていたスターシップも押した方向に動くはずです。

そこで、おもむろにAのボタンを押してみてください。スターシップからレーザーが発射されたはずです。

この発射されたレーザーが上を動いているスターキラーに運よく命中すれば、スターキラーは爆発します。何回もやってためしてみましょう。

スターキラーが爆発すると、もう一度やるかを聞いてきますのでもう１回やりたいときはY、やめたい場合はNを押して RETURN キーを押します。

コントローラーの状態を知るには

このプログラムを実行させてみてわかったと思いますが、今までのものと違い、コントローラーを使ってキャラクタを動かすことができます。

これはコントローラーの状態を調べ、それによって、キャラクタを　表示させているためです。コントローラーの状態を調べるにはＳＴＩＣＫとＳＴＲＩＧを使います。これは任意の変数にコントローラーの状態を数値の違いによって教えてくれるものです。

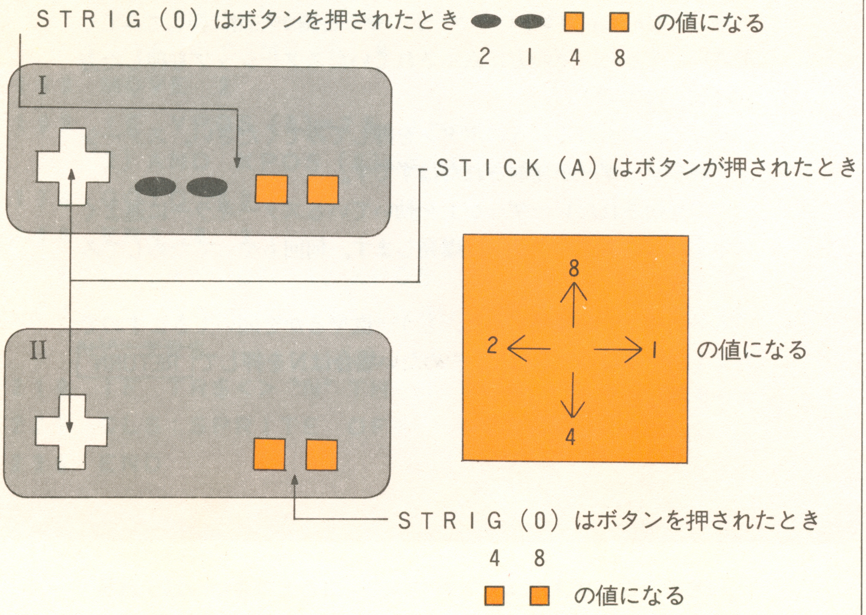
具体的な使い方は次の説明を見てください。

●ＳＴＩＣＫ　ＳＴＲＩＧの使い方

ＳＴＩＣＫ，（Ａ），ＳＴＲＩＧ（Ａ）

Ａ…コントローラーⅠのとき０，コントローラーⅡのとき１を入れる

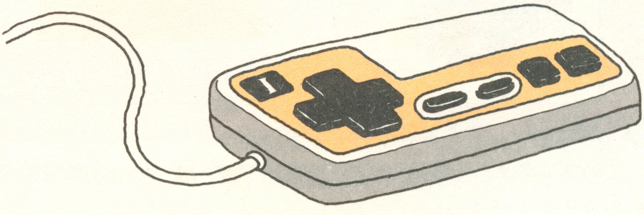
このＳＴＩＣＫ，ＳＴＲＩＧのときは、コントローラーの状態によって次のようになる



このプログラムの場合、90, 110行でＳＴＩＣＫ,140行でＳＴＲＩＧを使っています。90行ではコントローラーの✚が左側に押されていたときに変数ＳＸから１を引き、110行では右側に押されていたときにＳＸに１を加えます。ＳＸは80行でキャラクタを表示するＳＰＲＩＴＥの横方向の位置を設定するのに使っていますから、そのまま表示位置が変わるわけなのです。

```
80  SPRITE 0,SX,SY
90  IF STICK(0)=2 THEN SX=SX-1
100  IF SX<0 THEN SX=0
110  IF STICK(0)=1 THEN SX=SX+1
120  IF SX>220 THEN SX=220
130  IF MOVE(0) THEN 150
140  IF STRIG(0)=8 THEN POSITION 0,
SX+4,SY:MOVE 0
150  MOVE 2
```

さて、140行ではコントローラーのボタンＡが押されたときに、スターシップの位置からビームを動かすようにしています。ＰＯＳＩＴＩＯＮで、横方向の変数ＳＸに４を加えているのは、スターシップがビームの２倍の幅を持っているためです。スターシップの中央からビームが出ているように見せるために４を加えているわけなんですね。



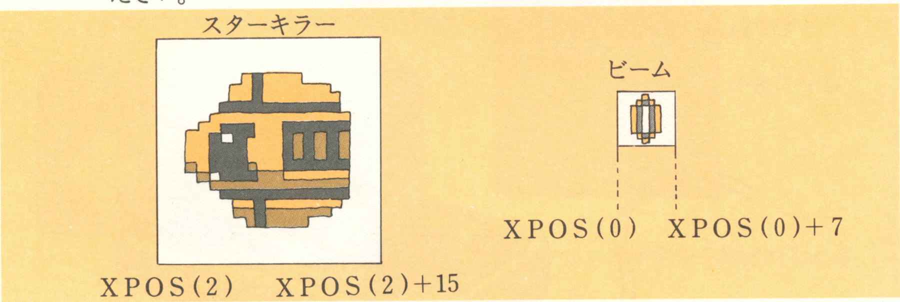
スプライトの衝突を知るには

このプログラムではスターシップから発射されたビームがスターキラーに命中するとスターキラーが爆発しますが、いったいどのような方法で行なっているのでしょうか。

ＭＳＸなどではスプライトが衝突したときにそのことを知らせる関数がありますが、ファミリーベーシックにはそのようなものはありません。したがって機械的にこの状態を知ることができませんのでプログラム的にその状態を知らせなくてはなりません。

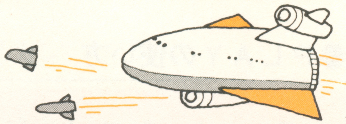
このプログラムでは160行でスターキラーとビームが重なったかどうかを調べています。２つのＩＦが使われていますが内容的にはそんなに難しいことをやっているわけではありません。まず最初にあるＹＰＯＳ（０）が28であるかというのは、ビームがスターキラーと縦方向が同じであるかを調べています。次のＩＦでは横方向が同じであるか、つまり、ＸＰＯＳ(２)で求めた値とそれに16を加えた値の間にあるかを調べています。アニメキャラクタを表示しましたが、ＸＰＯＳ、ＹＰＯＳでは求められる値はアニメキャラクタの左上のためです。スターキラーとビームが衝突しているかをプログラム的に知るためにはＩＦでＡＮＤを使ってこのように範囲を決めなくてはうまく調べることはできません。

当然ではありますがこの範囲を広くすれば衝突したときの判断は甘くなります。ＭＯＶＥを使ってキャラクタを動かす場合、速度が速ければ速いほどＸＰＯＳ、ＹＰＯＳで求められる値は飛び飛びになりますのでうまく判断できないときは範囲を大きめにとってみてください。



このプログラムは爆発のための判断やスターキラーを動かすものなので覚えてください。

サウンドをつけよう



```
140 IF STRIG(0)=8 THEN POSITION 0,
SX+4,SY:MOVE0:PLAY "T104C0"
195 PLAY "T100C9"
```

さきほどのコントローラーを使ったプログラムを次のように変えてみましょう。変更が終わりましたら、プログラムを実行してください。何も変わりませんか？　それではAボタンを押して、レーザーを発射させてみましょう。

おや？　ピッと音がしましたね。今度はなんとかして、スターキラーをやっつけてみましょう。スターキラーが爆発したときに、ブーと音がするはずです。

ゲームのプログラムは、そのほとんどにこのような効果音がつけられています。効果音の役割は背景の役割と非常によく似ていて、雰囲気を盛り上げたり、ゲームの進行のための判断要素として使ったりします。

ほとんどのゲームは雰囲気を盛り上げるために効果音を付けていますので、なくてもゲームはできます。ただ、やっぱりこれも「ないよりはあった方がいい」といった考えで、余裕があれば付けてみましょう。

さて、音を出すにはPLAYを使います。ファミリーベーシックでは3つの音を同時に出すことができますので、くふうすれば音楽を効果音的に付けることも可能です。

しかし、PLAYを使って音を出すと、その音が鳴り止むまで次の処理を行ないません。したがって、一度に出す音が長ければ長いほどプログラムの実行速度は遅くなります。

ですから、プログラムの実行速度をあまり変えないで効果音を付けたいのでしたら、上のような単純な音を出すだけにした方がいいでしょう。

●ＰＬＡＹの使い方

ＰＬＡＹ　"〔データ〕：〔データ〕：〔データ〕"

データ…音を出すためのデータ。次のような順番で指定するが省略も可能

①テンポ…Ｔ１～Ｔ８。数が大きいほど遅い
②デューティ効果…Ｙ１～Ｙ３。音色を指定するもの
③エンベロープ…Ｍ０～Ｍ１。Ｍ０でＯＦＦ。Ｍ１でＯＮ
④エンベロープの長さ…Ｖ０～Ｖ15。数が大きいほど長い
⑤オクターブ…Ｏ０～Ｏ５。数が多いほど高い
⑥音程…下のように対応。半音上げる場合は前に#をつける

音　程	ド	レ	ミ	ファ	ソ	ラ	シ
指定方法	C	D	E	F	G	A	B

⑦長さ…０～９。数が大きいほど長い
⑧休符…Ｒ０～Ｒ９。数が大きいほど長い
これらのデータを「：」で区切って使うと和音になる。
＊②～④の詳しい使い方はマニュアルの80ページを見てください

さて、このＰＬＡＹを使って作った音を効果音として使うわけです。しかし、プログラム上で毎回通る場所に長い効果音を入れると、今までゲームだったものが、速度が遅くなってしまったためにゲームにならなくなってしまうことがあります。

効果音は、短めのやつをＩＦ～ＴＨＥＮ～で条件判断によって処理されるところに入れた方がいいでしょう。

背景との合成とバックの色づけ

今度もさきほどのプログラムを消さずに入力してください。

```
15 VIEW:PALETB 0,33,80,21,7
```

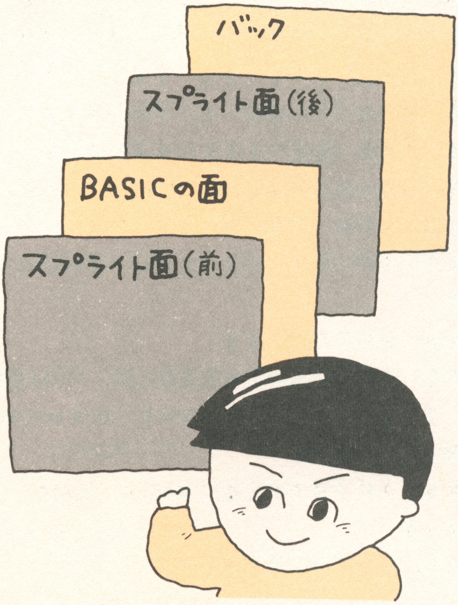
入力が終わったらまだ実行させないで、SYSTEMでGAME BASICモード画面にします。そこでBG GRAPHICを選びFILEした機能を使って、さきほどテープに記録したBACK1というBG GRAPHIC面を読み込んでください。そのままBG GRAPHIC面が残っている場合はテープから読み込む必要はありません。

さて背景の読み込みが終わったら再びBASICに戻り、プログラムを実行させてみましょう。するとBG GRAPHIC面がさっきのプログラムの画面と合成され、バックの色も黒から水色になったはずです。なんかちょっと見栄えがよくなりましたね。

実は上のプログラムでVIEWというのが背景を合成させるためのもので、PALETがバックの色を変えるためのものなんです。VIEWはBG GRAPHIC面で作った背景が合成される、というよりは、BASIC面にBG GRAPHIC面を写すものです。もちろんBG GRAPHIC面の色はBG GRAPHICで作ったものと同じになります。

さてPALETはさきほども出てきましたが、上の例でいえば33の部分が背景色になります。まだPALETの使い方がよくわからない人はバックの色を変えるときに上と同じものを33の部分だけ変えて使ってください。

これらの画面は右のような関係になっています。



迷路ゲームを作ってみよう

注：このプログラムはV.2.0以降で使えます

```
10 CLS:CGSET 1,1:SPRITE ON:VIEW:TI=20:
SC=0:LE=0
20 DEF SPRITE 0,(0,1,0,1,0)=CHR$(125)+
CHR$(124)+CHR$(127)+CHR$(126)
30 DEF SPRITE 1,(0,1,0,0,0)=CHR$(124)+
CHR$(125)+CHR$(126)+CHR$(127)
40 DEF SPRITE 2,(0,1,0,0,1)=CHR$(142)+
CHR$(143)+CHR$(140)+CHR$(141)
50 DEF SPRITE 3,(0,1,0,0,0)=CHR$(140)+
CHR$(141)+CHR$(142)+CHR$(143)
55 DEF MOVE (0)=SPRITE(0,7,TI,112,1,0)
56 LOCATE 8,0:PRINT"SCORE";SC;:LOCATE
18,0:PRINT"LEVEL";LE:LOCATE 11,21:PRIN
T"TIME":LOCATE 0,22:PRINT"* * * * * *
* * * * * * * * *"
60 CX=24:CY=38:S=0:TX=240:SP=2
62 PLAY"O3C5R5C5R5C5R5O5C6"
65 FOR I=0 TO 24:SPRITE S,I,CY:NEXT
66 POSITION 0,TX,192:MOVE 0
70 SX=(CX-12)/8+DX:SY=(CY-20)/8-1+DY
80 C$=SCR$(SX,SY)
190 SPRITE S,CX,CY
195 IF STICK(0)=1 THEN DX=2:DY=1:GOTO 2
00
196 GOTO 205
200 IF C$=CHR$(32) THEN SPRITE 1:SPRIT
E 2:SPRITE 3:S=0:CX=CX+SP:IF CX>216 TH
EN CX=216
205 IF STICK(0)=2 THEN DX=-1:DY=2:GOTO
 210
206 GOTO 215
```

```
210 IF C$=CHR$(32) THEN SPRITE 0: SPRIT
E 2:SPRITE 3:S=1:CX=CX-SP:IF CX<24 THE
N CX=24:IF CY>160 THEN 700
215 IF STICK(0)=4 THEN DX=1:DY=3:GOTO 2
20
216 GOTO 225
220 IF C$=CHR$(32) THEN SPRITE 0:SPRIT
E 1:SPRITE 3:S=2:CY=CY+SP:IF CY>168 TH
EN CY=168
225 IF STICK(0)=8 THEN DX=0:DY=0:GOTO 2
30
226 GOTO 500
230 IF C$=CHR$(32) THEN SPRITE 0: SPRIT
E 1:SPRITE 2:S=3:CY=CY-SP:IF CY<38 THE
N CY=38
500 IF MOVE(0)=0 THEN 800
510 GOTO 70
700 FOR I=0 TO 24:SPRITE S,CX-I,CY:NEX
T:CUT 0
710 LOCATE 7,9:PRINT"GOAL IN !!"
720 LOCATE 3,11:PRINT"CHALLENGE NEXT L
EVEL !!"
730 TI=TI-2:IF TI<1 THEN TI=1
740 SC=SC+XPOS(0):LOCATE 8,0:PRINT"SCO
RE";SC
750 LE=LE+1
760 PAUSE 200:SPRITE S:ERA 0:VIEW:GOTO
 55
800 FOR I=0 TO 10:LOCATE 9,9:PRINT"GAM
E OVER":PAUSE 20:LOCATE 9,9:PRINT"
      ":PAUSE 20:NEXT
810 LOCATE 7,9:INPUT"TRY AGAIN ";TR$:
IF TR$<>"N" THEN RUN
```

```
820 CLS:CGSET 1,0:SPRITE OFF
830 END
```

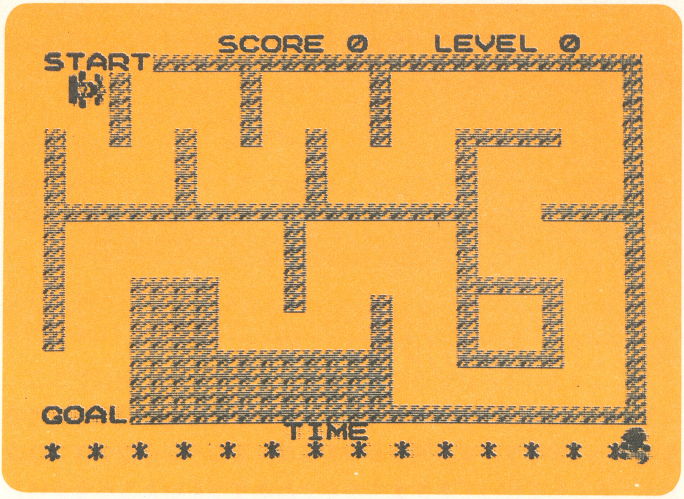
順番はどちらでもかまいませんが、プログラムと背景の両方を入力してください。もちろん背景はBG GRAPHICを使って入力しますがキャラクタはいずれのものを使ってもかまいません。プログラムではキャラクタが何であるかを調べてはいないからです。これについては後で説明しますが、ここでは何も考えずに下と同じものを作ってください。使っているキャラクタはKグループの３番で配色番号は２です。

プログラムは長いですが間違えずに正確に入力してください。「：」を使って１つの行番号でいくつものことをやっていますが、これはメモリを節約するためです。
これを全部１つずつ行番号を付けていったら、ファミリーベーシックのメモリが足りなくなってしまうばかりか、プログラムもかなりわかりづらいものになってしまうはずです。極端に多用しないのでしたら使うことをお勧めします。

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1	S	T	A	R	T	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31
2				G31						G31						G31												G31
3				G31						G31						G31												G31
4				G31						G31						G31												G31
5	G31			G31			G31			G31			G31			G31				G31	G31	G31	G31	G31				G31
6	G31						G31						G31							G31								G31
7	G31						G31						G31							G31								G31
8	G31						G31						G31							G31								G31
9	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31								G31
10	G31											G31								G31				G31	G31	G31	G31	G31
11	G31											G31								G31								G31
12	G31											G31								G31								G31
13	G31				G31	G31	G31	G31				G31				G31				G31	G31	G31	G31	G31				G31
14	G31				G31							G31				G31				G31								G31
15	G31				G31							G31				G31				G31								G31
16	G31				G31				G31	G31	G31	G31				G31				G31								G31
17					G31											G31								G31	G31	G31	G31	G31
18					G31											G31								G31	G31	G31	G31	G31
19					G31											G31								G31	G32	G31	G31	G31
20	G	O	A	L	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31	G31

ゲームの内容

さて、プログラムと背景の入力が終わったらプログラムを実行させてみましょう。次のような画面になり、エラーの表示が出なければいいのですが、もしそうならないときは、ＬＩＳＴを使ってプログラムを画面に表示させ、入力間違いをさがしてください。



このプログラムはゲームになっています。今、画面の左上にＳＴＡＲＴと書かれたところに車が出てきました。この車をコントローラーＩを使って操作し、画面の左下のＧＯＡＬまで動かしてください。

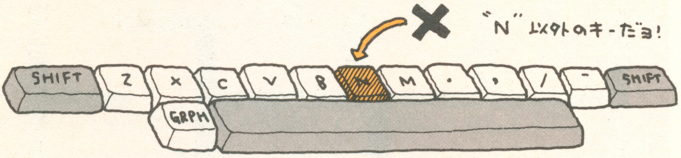
画面の下を動いているマリオは時間の経過を表しています。最初はいちばん右にいますがこれがいちばん左まで動いてしまうと、ゲームオーバーとなってしまうのでそれまでにゴールに入らなくてはなりません。

車をゴールに入れると残りの時間がスコアに加算されますので速くゴールに入った方が高い得点を取ることができます。

時間内に車をゴールに入れると再びスタートのところに車が出てきます。また車をゴールまで動かせばいいのですが、今度はマリオの動く速度が少し速くなっています。つまり、時間が短くなったわけですね。今度もマリオが画面の左端に行くまでに車をゴールに入れればいいわけです。得点の加算方法も最初のときと同じですから速くゴールに入った方が高い得点を取れるわけです。

車がゴールに入ると再び時間が短くなり車がスタート位置から出てきます。マリオが左端に行くまでに車をゴールさせればゲームはずっと続きますが、そのたびに時間は短くなっていきます。

マリオが左端に行くとゲームオーバーとなり、もう一度ゲームをやるのか聞いてきます。ゲームをやる場合はN以外のキーを押してください。するとまたゲームが始まります。Nを押せばプログラムは終了します。



●プログラムの内容を見てみよう

今までのプログラムはファミリーベーシックの命令を覚えるためのものだったのでサンプル的なものでした。ところがこのプログラムはゲームとして作ってありますので、わかりづらいところも多いと思います。

そこでこのプログラムを少しずつ分解して説明してみましょう。

①初期設定

```
10 CLS:CGSET 1,1:SPRITE ON:VIEW:TI=20:
SC=0:LE=0
```

ここでは画面の消去、パレットの設定、スプライト面の表示、1部変数の初期値を設定しています。変数は次のように決めてあります。

TI………マリオの速度の値
SC………スコアの値
LE………レベルの値

マリオの速度というのはDEF MOVEでマリオを設定するときの速さです。マリオは経過時間を表しますが、このTIの値を小

さくすればそれだけ時間が短くなるわけです。

②キャラクタの設定

```
20 DEF SPRITE 0,(0,1,0,1,0)=CHR$(125)+
CHR$(124)+CHR$(127)+CHR$(126)
30 DEF SPRITE 1,(0,1,0,0,0)=CHR$(124)+
CHR$(125)+CHR$(126)+CHR$(127)
40 DEF SPRITE 2,(0,1,0,0,1)=CHR$(142)+
CHR$(143)+CHR$(140)+CHR$(141)
50 DEF SPRITE 3,(0,1,0,0,0)=CHR$(140)+
CHR$(141)+CHR$(142)+CHR$(143)
55 DEF MOVE (0)=SPRITE (0,7,TI,112,1,0)
```

ここでは使うアニメキャラクタの設定を行なっています。DEF SPRITEで設定している4つのアニメキャラクタは車でDEF MOVEで設定しているアニメキャラクタはマリオです。4つ設定したスプライトは次のようになっています。

スプライト番号0……右向きの車
スプライト番号1……左向きの車
スプライト番号2……下向きの車
スプライト番号3……上向きの車

車を4方向について設定したのはもちろんコントローラーを使って車を4方向に動かせるようにプログラムを作るためです。

車を動かすのにDEF MOVEを使わずにDEF SPRITEで行なっているのはMOVEでキャラクタを動かしているときにXPOS, YPOSの値が正確に出てこないためです。後で説明しますが、このプログラムは動いている車の前方についてどんなキャラクタがあるかを調べています。このときにMOVEで動作させたキャラクタをXPOS YPOSで位置を求めてから行なうとどうしても正確に位置を求められないことがあります。

したがって前方の物があるかどうかを確認できないことがあるわけなのです。

DEF　SPRITEでアニメキャラクタを設定し、SPRITEで動かすには位置をこちらで設定するわけですからこのようなことはありえないわけです。

マリオのみをDEF　MOVEを使って設定しているのは、DEF　MOVEを使って動かしたキャラクタはプログラムに関係なく速度が一定であるためです。これが時間経過を表すわけですから一定でなくてはこまるわけです。

③初期設定　その2

```
56 LOCATE 8,0:PRINT "SCORE";SC;:LOCATE 18,0:PRINT "LEVEL";LE:LOCATE 11,21:PRINT "TIME":LOCATE 0,22:PRINT "* * * * * * * * * * * * * * *"
60 CX=24:CY=38:S=0:TX=240:SP=2
```

ここでは画面表示と残りの変数に初期値を設定しています。画面に表示しているのはスコア、レベル、マリオが動く場所のスケールです。マリオは時間を示すものですが、スケールがあったほうがわかりやすいと思い付けてみました。マリオはBASICの面より後に表示されますのでわかりやすいはずです。

さて、変数ですが、次のようになっています。

CX………車の横方向位置
CY………車の縦方向位置
S…………現在使用しているスプライト番号
TX………マリオの横方向位置
SP………車の動く量

このプログラムでは車の動く量を2にしていますが、今のままでは遅く感じるのでしたら、数を増せば動きが速くなります。

しかし、数を増すとだんだん動き方がぎこちなくなってきますので4ぐらいが限界でしょう。

④車のスタート

```
62 PLAY "O3C5R5C5R5C5R5O5C6"
65 FOR I=0 TO 24:SPRITE S,I,CY:NEXT
66 POSITION 0,TX,192:MOVE 0
```

ここでは車をスタートさせるときの音、車のスタート、マリオの動作を行なっています。プログラムを実行させた人はわかっていると思いますが、「プ，プ，プ，ピーン」という音を出して車がスタート位置に飛び出してくるところをここで行なっています。

ファミリーベーシックでは音楽の演奏が終わるまで次の命令を実行しませんのでウェイトを取る必要はありません。

```
70 SX=(CX-12)/8+DX:SY=(CY-20)/8-1+DY
80 C$=SCR$(SX,SY)
```

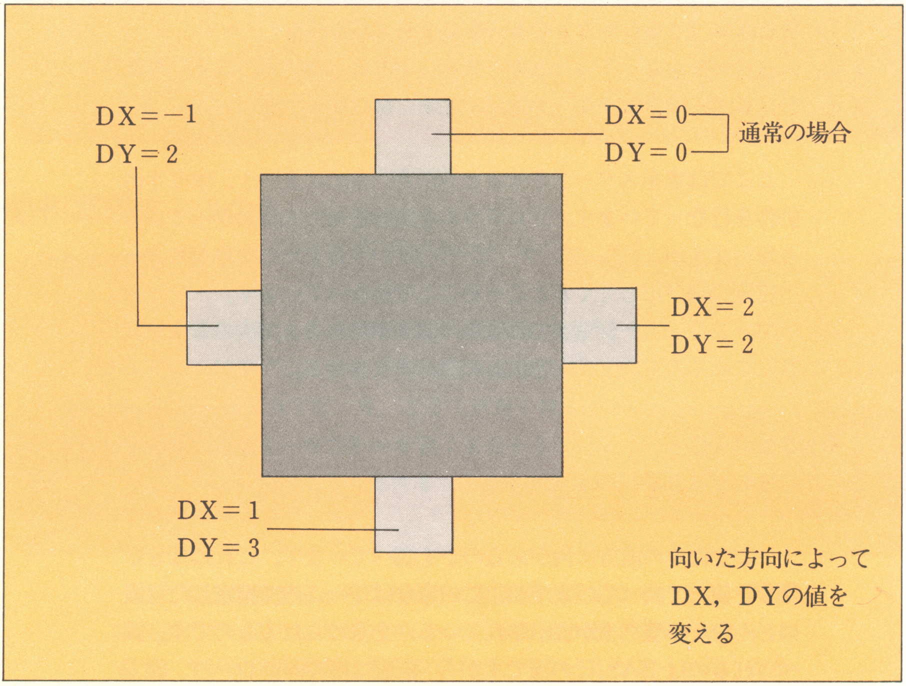
ここでは車の前方に何があるのかを調べています。これにはＳＣＲ＄を使って調べていますが、この命令はかっこ内で指定した位置にどんなキャラクタが表示されているかを求められるものです。調べられるのは BASIC の面ですから、位置は横方向が０～27，縦方向が０～23の範囲です。

車はスプライトを使って表示していますので、位置は横方向が０～255，縦方向が０～239となります。これでわかるように、車の前方を調べるには値の変換をしなければなりません。ＳＸ，ＳＹに前輪の位置を BASIC の面に変換させた値を代入しています。

ここでＤＸ，ＤＹを加えているのは車の向いている方向によって調べる位置が変わるためです。車の向きが変わっても車の前方には変わりませんが、実際は表示されている車の上下左右になりますので、そのための修正を行なわなければなりません。

毎回全方向について調べれば問題ありませんが、プログラムの実行速度が遅くなっています。また、車の前方向には最高で２つのキャラクタがあることになりますが、ここでは１つしかしらべていま

せん。これもプログラムの実行速度を速くするためです。詳しくは次の図を見てください。



⑤コントローラーの読み取り、車の表示

```
195 IF STICK(0)=1 THEN DX=2:DY=1:GOTO 2
 00
196 GOTO 205
200 IF C$=CHR$(32) THEN SPRITE 1:SPRIT
E 2:SPRITE 3:S=0:CX=CX+SP:IF CX>216 TH
EN CX=216
205 IF STICK(0)=2 THEN DX=-1:DY=2:GOTO
 210
206 GOTO 215
```

```
210 IF C$=CHR$(32) THEN SPRITE 0:SPRIT
E 2:SPRITE 3:S=1:CX=CX-SP:IF CX<24 THE
N CX=24:IF CY>160 THEN 700
215 IF STICK(0)=4 THEN DX=1:DY=3:GOTO
220
216 GOTO 225
220 IF C$=CHR$(32) THEN SPRITE 0:SPRIT
E 1:SPRITE 3:S=2: CY=CY+SP:IF CY>168
THEN CY=168
225 IF STICK(0)=8 THEN DX=0:DY=0:GOTO
230
226 GOTO 500
230 IF C$=CHR$(32) THEN SPRITE 0:SPRIT
E 1:SPRITE 2:S=3:CY=CY-SP:IF CY<38 THE
N CY=38
```

ここではコントローラーの状態を読みとり、それによってさまざまな値を変化させています。いろいろなことをやっているのでわかりづらいかもしれませんが、基本的には2つの行ずつで成り立っていて、やっていることはほとんど同じで次のようになっています。

①コントローラーの✚の状態を調べる
②DX，DYの値を変える
③車の前方がスペース〔CHR＄(32)〕かを調べる
④進行方向以外のスプライトを消す
⑤Sの値を現在の方向に変える
⑥CX（もしくはCY）の値を変化させる
⑦CX（CY）の値が背景の位置よりオーバーしていないかを調べる

車の前方のキャラクタを代入したC＄はCHR＄(32)であるスペースかどうかを調べているだけです。したがって、背景で使うキャラクタは何でもいいだけではなく、コース自体も今のものと違っていてもかまわないわけです。プログラムの内容がわかったら自分でコースを作るのもいいでしょう。

さて、ここではこの他にゴールに入ったときの判断を行なっています。210行を見てください。変化させている値はCXだけですがCYの値の状態も調べています。これでゴールの位置に車がいるのかを調べているわけなのです。

⑥ゲームオーバーの判断

```
500 IF MOVE(0)=0 THEN 800
510 GOTO 70
```

ゲームオーバーはマリオが左端まで動いたことで処理されなければなりませんのでMOVE(0)を使ってマリオが動作中かどうかを調べることで行ないます。もし停止しているのなら、ゲームオーバーの処理を行なう800行へジャンプします。

510行でゲーム中の処理はすべて終りとなり、再び画面を読み取る700行へ流れを移しています。

⑦ゴールインの処理

```
700 FOR I=0 TO 24:SPRITE S,CX-I,CY:NEXT:CUT 0
710 LOCATE 7,9:PRINT"GOAL IN !!"
720 LOCATE 3,11:PRINT"CHALLENGE NEXT LEVEL !!"
730 TI=TI-2:IF TI<1 THEN TI=1
740 SC=SC+XPOS(0):LOCATE 8,0:PRINT"SCORE";SC
750 LE=LE+1
760 PAUSE 200:SPRITE S:ERA 0:VIEW:GOTO 55
```

210行で車がゴールに入ったと判断されたとき、この部分が実行されます。ここでは車をゴールの中に入れ「GOAL IN!!」

「CHALLENGE　NEXT　LEVEL！！」の表示を行なった後、スコア、レベルアップ、新しい面の画面表示を行なっています。

マリオを動かす速度、つまり時間の経過は2ずつ少なくしていますが、1より小さい値を入れてしまうと、DEF　MOVEのところでエラーになってしまいますので、TIが1より小さくなってしまったらTIを1にするようにしています。

スコアの加算はマリオの横方向位置であるXPOS（0）で行なっています。したがって、マリオの動いた距離が少なければ少ないほど得点は高くなります。760行は次のレベルになるときの画面の描き直しを行なっています。

⑧ゲームオーバーの処理

```
800 FOR I=0 TO 10:LOCATE 9,9:PRINT"GA
ME OVER":PAUSE 20:LOCATE 9,9:PRINT"
          ":PAUSE 20:NEXT
810 LOCATE  7,9:INPUT"TRY AGAIN ";TR$:
IF TR$<>"N" THEN RUN
820 CLS:CGSET 1,0:SPRITE OFF
830 END
```

500行でマリオの動きが終了し、ゲームオーバーとなったときこの部分が実行されます。ここでは「GAME　OVER」の表示と、リプレイをするのかをたずねます。

800行の「GAME　OVER」を表示させるところで、FOR～TO～　NEXTやPAUSEなどがあるのは、「GAME　OVER」の字を点滅させて見せるためです。

810行では「TRY　AGAIN」の文字を表示させ、TR$にキー入力文字を代入していますが、これがNの場合のみリプレイになり、他の場合は820行を実行し、プログラムが終了します。Yを入力するとリプレイになる方法もありますが、こちらの方が便利でしょう。

他の機種用に作られたプログラムをファミリーコンピュータで動かすには?

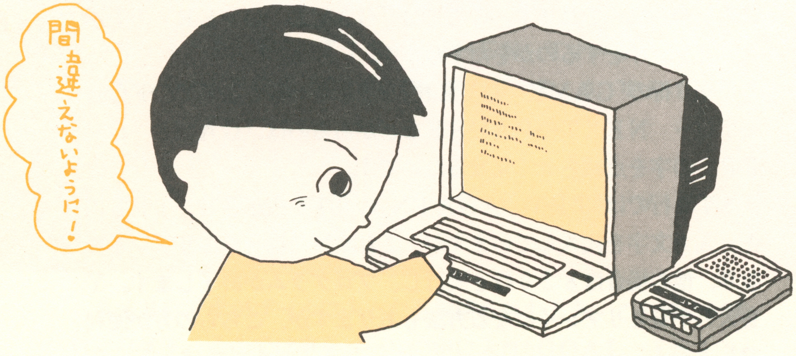
BASIC言語が使えるパーソナルコンピュータはかなりたくさんあります。PC−8801やFM−7、X1などが有名ですが、ファミリーコンピュータもファミリーベーシックでBASIC言語が使えるパーソナルコンピュータになることは周知のことだと思います。

これらのパーソナルコンピュータには、それぞれの機種用に作られたBASICのゲームなどのプログラムがよく本に載っていますが、PC−8801やFM−7、X1などと比較するとファミリーコンピュータ用は少数派と言えるでしょう。

そこで思いつくのが、「他の機種用に書かれたBASICのプログラムをファミリーコンピュータで使えないのか」ということです。同じBASICで書かれているのだから使えそうだと考えるのはむしろ当然のことです。

ところが結論から言えば9割がたは無理なのです。各機種用に書かれたプログラムはその機種の命令を使っていますから、よほどのことがないかぎり、そのままプログラムを入力しても動くことはありません。移植という方法もありますが、どちらのパソコンについても詳しく使っていなくてはできませんので、まず無理でしょう。

他機種用のプログラムを使いたいと思ったときは、アイディアやプログラムの流れなどを参考にファミリーコンピュータで作りなおすのが、最も速い方法といえるでしょう。



FAMILY COMPUTER
FAMILY BASIC
PERFECT BOOK

PART 5

ファミリーベーシック
コマンド早見表

▲演算子

●算術演算子

＋	加算	A ＋ B	AにBを加える
−	減算	A − B	AからBを引く
＊	乗算	A ＊ B	AとBをかける
／	除算	A ／ B	AをBで割る
MOD	剰余	A MOD B	AをBで割った余り

●関係演算子

＝	A ＝ B	AとBが等しい
＜ ＞	A ＜ ＞ B	AとBが等しくない
＞	A ＞ B	AはBより大きい
＜	A ＜ B	AはBより小さい
＞ ＝	A ＞ ＝ B	AはB以上
＜ ＝	A ＜ ＝ B	AはB以下

●論理演算子

NOT	否定
AND	論理積
OR	論理和
XOR	排他的論理和

NOT　A

A	NOT　A
0	1
1	0

A AND B,　A OR B,　A XOR B,

A	B	A AND B	A OR B	A XOR B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

▲コマンド

・CLEAR

働き……BASICで使用するメモリの最上アドレスを決める

文法……CLEAR　A

A…77FFH以下のアドレス値

・NEW

働き……メモリ上のプログラムをすべて消す

文法……NEW

・LIST

働き……メモリ上のプログラムを画面に表示させる

文法……LIST〔＜行番号＞　＜－行番号＞〕

・RUN

働き……メモリ上のプログラムを実行させる

文法……RUN〔行番号〕

・CONT

働き……プログラムの再開

文法……CONT

・LOAD

働き……カセットテープに記録したプログラムをメモリ上に読み込む

文法……LOAD〔"ファイル名"〕

・SAVE

働き……メモリ上のプログラムをカセットテープに記録する

文法……SAVE〔"ファイル名"〕

・LOAD？

働き……カセットテープに記録したプログラムが正しく記録されているか確かめる

文法……LOAD？〔″ファイル名″〕

▲一般ステートメント

・PRINT

働き……画面に表示を行なう

文法……PRINT〔画面に表示させたいもの〕

・INPUT

働き……キーボードから入力したものを変数にとりこむ

文法……INPUT〔″文字″〕〔；か，〕変数〔，変数・・〕

・LINPUT

働き……キーボードから入力した文字を３１文字まで文字変数にとりこむ

文法……LINPUT〔″文字″〕〔；か，〕文字変数

・CLEAR

働き……変数、配列をすべてクリアする

文法……CLEAR

・DIM

働き……配列宣言をする

文法……DIM　配列名〔，配列名・・〕

・GOTO

働き……指定した行番号へプログラムの流れを変える

文法……GOTO　行番号

・GOSUB

働き……プログラム内のサブルーチンへプログラムの流れを変える
文法……GOSUB　行番号

・RETURN

働き……サブルーチンからGOSUBの後ろへ戻る
文法……RETURN〔行番号〕

・IF～THEN

働き……条件判断を行い、THEN以降を実行する
文法……IF　条件式　THEN　実行内容か行番号

・FOR～TO～STEP, NEXT

働き……指定された分だけ　FOR～NEXT間の実行を繰り返す
文法……FOR　a＝b　TO　c〔STEP　d〕
NEXT
a…数値変数、b…初期値、c…終了値、d…増分

・ON

働き……式の値が1，2，3，4，5のとき指定した行番号へプログラムの流れを変える
文法……ON　式　GOTOかGOSUBかRETURNかRESTORE　行番号〔，行番号・・〕

・STOP

働き……プログラムの実行を中止する
文法……STOP

・END

働き……プログラムの終了を宣言する
文法……END

・SWAP

働き……指定した2つの変数の内容を交換する

文法……SWAP　変数①，変数②

・REM

働き……プログラム中にコメントなどの非実行文を入れる

文法……REM〔非実行文〕

・READ

働き……DATAにあるデータを指定した変数にとりこむ

文法……READ　変数〔，変数・・〕

・DATA

働き……READで変数にとりくむデータを用意する非実行文

文法……DATA　定数〔，定数・・〕

・RESTORE

働き……指定した行番号にあるDATAをREADでとりこませる

文法……RESTORE〔行番号〕

・CALL

働き……指定したアドレスのマシン語サブルーチンを呼び出す

文法……CALL　アドレス

・POKE

働き……メモリ上に1バイトずつデータを書き込む

文法……POKE　アドレス，データ〔，データ・・〕

▲画面制御ステートメント

・LOCATE

働き……カーソルを指定した位置へ移動させる

文法……LOCATE　X，Y

X…横方向の位置で０～２７，Y…縦方向の位置で０～２３

・COLOR

働き……指定したエリアの配色番号を変える

文法……COLOR　X，Y，n

X…横方向の位置で０～２７，Y…縦方向の位置で０～２３

n…配色番号で０～３

・CGEN

働き……バックグラウンド面、スプライト面で使うキャラクタを割り当てる

文法……CGEN　n

n…割り当てるくみあわせで０～３

・CLS

働き……バックグラウンド面をすべて消す

文法……CLS

・CGSET

働き……バックグラウンド面、スプライト面で使うパレットコードを指定する

文法……CGSET　a，b

a…バックグラウンド用パレットコードで０～１

b…スプライト用パレットコードで０～２

・PALET

働き……バックドロップ面、配色番号の色を変える

文法……PALET　BかS　n，a，b，c，d

B…バックドロップ面用、S…スプライト用、n…他配色番号で０～３、

a…バックドロップ面の色で０～６０（配色番号が０のとき有効）

b,c,d…配色番号の色に対応するもので０～６０

▲MOVEコマンド

・DEF　MOVE

働き……アニメキャラクタの動きを定義する

文法……DEF　MOVE(n)＝SPRITE(A,B,C,D,E,F)

n…アニメキャラクタの動作番号で０～７，A…アニメキャラクタの種類で０～１５、B…動作の方向で０～８、C…動作の速さで１～２５５，D…全移動量で１～２５５，E…表示優先度で０か１

・MOVE

働き……DEF　MOVEで定義したアニメキャラクタを動作させる

文法……MOVE　動作番号〔，動作番号・・〕

・CUT

働き……MOVEで動作させたアニメキャラクタを止める

文法……CUT　動作番号〔，動作番号・・〕

・ERA

働き……MOVEで動作させたアニメキャラクタを止めて消す
文法……ERA　動作番号〔, 動作番号・・〕

・POSITION
働き……MOVEで動作させたアニメキャラクタの初期位置を決める
文法……POSITION　　動作番号, X, Y
X…横方向の位置で0～255、Y…縦方向の位置で0～255

・XPOS
働き……MOVEで動作させたアニメキャラクタの横方向の位置を求める
文法……XPOS（動作番号）

・YPOS
働き……MOVEで動作させたアニメキャラクタの縦方向の位置を求める
文法……YPOS（動作番号）

・MOVE（動作番号）
働き……MOVEで動作させたアニメキャラクタの移動中かどうかを調べる
文法……MOVE（動作番号）

▲特殊ステートメント

・KEY
働き……ファンクションキーを定義する
文法……KEY　番号（1～8), 〃文字〃

・KEYLIST

働き……ファンクションキーの定義内容を表示する

文法……KEY　LIST

・PAUSE

働き……プログラムの実行を一時停止させる

文法……PAUSE〔n〕

n…1時停止させる長さで0～32767

・SYSTEM

働き……BASICからGAME　BASICモード画面に戻る

文法……SYSTEM

・VIEW

働き……BG　GRAPHIC面をBASIC面に写す

文法……VIEW

▲サウンド制御ステートメント

・BEEP

働き……ピーッという音を出す

文法……BEEP

・PLAY

働き……指定したデータ通りに音楽を演奏する

文法……PLAY〝演奏データ〃

▲関数

＊数値関数＊

・ABS

働き……数式の絶対値を求める

文法……ＡＢＳ（数式）

・SGN

働き……数式の符号を求める

文法……ＳＧＮ（数式）

・RND

働き……数式の値未満の乱数を求める

文法……ＲＮＤ（数式）

＊文字関数＊

・ASC

働き……文字をキャラクタコードに交換する

文法……ＡＳＣ（文字列）

・CHR$

働き……キャラクタコードを文字に交換する

文法……ＣＨＲ＄(数式)

・VAL

働き……数字の文字列を数値に交換する

文法……ＶＡＬ（文字列）



・ＳＴＲ＄
働き……数式を数字の文字列に交換する
文法……ＳＴＲ＄(数式)

・ＨＥＸ＄
働き……１０進の数式を１６進数の文字列に交換する
文法……ＨＥＸ＄(数式)

・ＬＥＦＴ＄
働き……指定した分だけ文字列の左側から文字を取り出す
文法……ＬＥＦＴ＄(文字列、取り出す文字数)

・ＲＩＧＨＴ＄
働き……指定した分だけ文字列の右側から文字を取り出す
文法……ＲＩＧＨＴ＄(文字列、取り出す文字数)

・ＭＩＤ＄
働き……指定した位置から指定した分だけ文字列から文字を取り出す
文法……ＭＩＤ＄(文字列、開始位置、取り出す文字数)

＊特殊関数＊

・ＰＥＥＫ
働き……指定したアドレスからデータを取り出す
文法……ＰＥＥＫ（アドレス)

・ＰＯＳ
働き……カーソルの横方向の位置を求める
文法……ＰＯＳ（０）

・ＦＲＥ
働き……ＢＡＳＩＣのプログラムで使っていないメモリのバイト数を求める
文法……ＦＲＥ

・STICK

働き……コントローラーの方向ボタンの状態を求める

文法……STICK (n)

n…0のときコントローラーIで、1のときコントローラーII

・STRIG

働き……コントローラーのトリガーボタンの状態を求める

文法……STRIG (n)

n…0のときコントローラーIで、1のときコントローラーII

・CSRLIN

働き……カーソルの縦方向の位置を求める

文法……CSRLIN

・SCR$

働き……バックグラウンド面の状態を求める

文法……SCR$(X, Y, S)

X…横方向の位置で0～27, Y…縦方向の位置で0～23, S…配色番号を求める場合は1で, 求めない場合は0

＊入力用文字関数＊

・INKEY$

働き……キーボードから入力された1文字を求める

文法……INKEY$〔(0)〕



▲スプライト制御文ステートメント

・DEF　SPRITE

働き……スプライトにアニメキャラクタを定義する

文法……DEF　SPRITE　n，(A，B，C，D，E)＝文字列

n…スプライト番号で０～７，A…配色番号で０～３，
B…キャラクタの大きさで０か１（０のとき８×８ドット、１のとき１６×１６ドット）、
C…表示優先度で０か１（０のときバックグラウンド面より前、１のとき後）
D…X軸方向反転表示で０か１（０のときキャラクタテーブルと同じ、１のとき反転）、
E…Y軸方向反転表示で０か１
（０のときキャラクタテーブルと同じ、１のとき反転）

・SPRITE

働き……DEF　SPRITEで定義したアニメキャラクタを指定した位置に表示と消去する

文法……SPRITE　スプライト番号，X，Y

X…横方向の位置で０～２５５、Y…縦方向の位置で０～２５５

・SPRITE　ON

働き……スプライト面を表示させる

文法……SPRITE　ON

・SPRITE　OFF

働き……スプライト面を消す

文法……SPRITE　OFF

プログラムを作ってメモリがたりなくなってしまったら

ファミリーベーシックのV.2.0はメモリ容量が2Kバイトしかありませんので、プログラムを作っているとメモリがたりなくなってしまうことがしばしばあります。たいていのパソコンは16～64Kバイトもメモリがありますので、よほど大きなプログラムを作らない限り、メモリがたりなくなることはありませんが、ファミリーコンピュータの場合はちょっと長めのものを作ると、すぐにメモリがたりなくなってしまいます。

このようなときには、プログラムを入力した時点では、なにも起こりませんが、RUNで実行させたときに、?　OM　ERRORの表示が出ます。エラーが出てしまったときはプログラムを小さくするしかありませんが、自分で考えていたものが思うようにできないのはつらいものです。

そこでプログラムを小さくしないで、メモリを節約する方法をお教えしましょう。まず、REMを使わないということです。REMを使ってプログラム中にコメントを入れるとわかりやすいですが、プログラム的には全くムダなので、紙などに行番号を書いておいて代用します。

次に配列変数をできるだけ使わないようにします。これは変数エリアが少なくてすむようにするためです。

最後はFOR～NEXT、GOSUB～RETURNをできる限り少なくします。これらは、飛び先を覚えておくためにメモリを使いますので、GOTOやIF～THEN～で代用すれば、多少メモリが節約できます。

この他にマルチステートメントを多用する、スペースを全部取ってしまうというのもありますが、修正が大変になりますので、ほどほどにしましょう。

FAMILY COMPUTER FAMILY BASIC PERFECT BOOK

PART 6

ファミリーベーシック ゲームプログラム

ここには、ファミリーベーシックの特長を生かしたゲームプログラムがたくさん載っています。

プログラムは印刷の文字で書かれていますので、実際の画面に表示されるものと桁数が異なっています。入力中に行か変わってしまっても全く関係ありませんので気にせずに続けてください。また、数字の1と英文字のI、数字の0と英文字のOは非常に間違えやすいので注意しましょう。

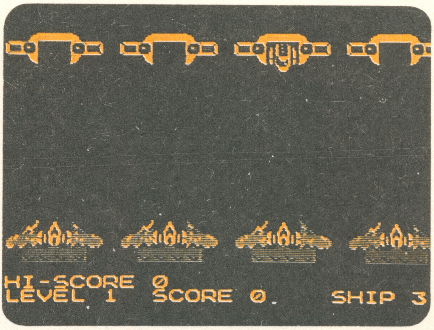
BG GRAPHIC で作った背景面のデータはマニュアルと同じ法則で書いてあります。たとえば、G10とあれば、マニュアルの93ページや113ページにあるキャラクタテーブル、BのGグループの1のキャラクタで、配列番号0を使う、ということです。また、これ以外の表記で書かれているところはCHARモードでキーボードから直接文字を入力するということです。実際にどのような画面になるのかは、写真を見て参考にしてください。配色番号はプログラムとかかわりが深いものがありますので、必ずデータと同じものを使ってください。

さて、プログラムと背景の入力が終わったら、実行させてみます。正しく動かなかったときは STOP キーを押してプログラムを止め、画面を消してから修正してください。

修正が終わったら、テープに記録しておきます。順番はどちらからでもかまいませんが、背景を先に、プログラムを後にすると、今度ロードするときに楽です。ファイトネームはわかりやすいようにしっかり入れておきましょう。

1 ディフェンド・ベース

ゲームの遊び方



4つある基地のスターシップをUFOから守るゲームです。基地にあるスターシップや、UFOから出現するスターキラーは飛行能力はありませんが、ビームを発射することができます。スターキラーより速くビームを発射すればスターシップを守ることができます。基地にあるスターシップからビームを発射するには[Z][X][,][／]のキーで、それぞれの位置のスターシップに対応しています。

UFOから出現するスターキラーは、ときどきスターシップに姿を変えているときがあります。出現したスターシップはビームを発射してきませんが、これに向かってビームを発射すると、予備のスターシップが1機減ります。スターシップに基地のスターシップを攻撃されても、予備のスターシップが1機減ります。予備のスターシップが無くなるとゲームオーバーとなりますので、慎重にビームを発射してください。

UFOからスターキラーやスターシップが10機出現するとレベルが上り、スターキラーがビームを発射するまでの時間が短くなります。以後、同じ調子でレベルはどんどん上っていきます。

最初にプログラムを実行させると「PUSH ANY KEY TO START!!」と表示されます。このとき何かキーを押せばゲームがスタートするようになっています。

また、ゲームオーバーになってしまったときにN以外のキーを押してからリターンキーを押すと、再びゲームが始まります。

ゲーム中のハイスコアが記録できるようになっていますので、家族や友人と反射神経と集中力を競い合ってみましょう。

プログラムの説明

10行	画面、変数の初期設定
20〜40行	基地のスターシップの設定
50行	爆発のキャラクタ設定
60行	ＵＦＯから出るスターキラーの設定
70行	ＵＦＯから出るスターシップの設定
80行	ビームの設定
85〜87行	ゲームスタート前のウェイト
90〜120行	基地のスターシップを表示
130行	レベルクリアのための初期値を設定
135〜140行	ハイスコア，スコア，レベル，予備のスターシップなどの数値を表示
150行	ＵＦＯから出現するキャラクタを決定
155行	レベルクリアの判断
160行	ＵＦＯから出現するキャラクタが何かを判断
170行	ＵＦＯからキャラクタを出現させる
190〜220行	キー入力の判断
230〜235行	タイマによってスターキラーからビームを発射させるための判断
240〜260行	スターシップからビームを発射させる
270〜300行	ビームがスターキラーに命中したかの判断
310〜315行	ＵＦＯから出現したキャラクタの爆発
320行	ＵＦＯから出現したキャラクタがスターシップのときの処理
330〜340行	ＵＦＯから出現したキャラクタがスターキラーのときの処理
400〜440行	レベルクリアのときの処理
500〜530行	スターキラーのビーム発射
540〜545行	基地のスターシップの爆発
550〜570行	予備のスターシップを減らし、ゲームオーバーの判断
600〜650行	ゲームオーバーの処理とリプレイの処理

●BASICリスト

```
10 CLS:CGSET 1,1:SPRITE ON:VIEW:SI=3:T
1=100:T2=100:SC=0:LE=1:PLAY "T1"
20 FOR I=0 TO 3
30 DEF MOVE(I)=SPRITE(9,0,1,1,1,0)
40 NEXT
50 DEF MOVE(4)=SPRITE(10,0,2,4,0,3)
60 DEF SPRITE0,(1,1,1,0,1)=CHR$(162)+CHR$(1
63)+CHR$(160)+CHR$(161)
70 DEF SPRITE1,(0,1,1,0,1)=CHR$(174)+CHR$(1
75)+CHR$(172)+CHR$(173)
80 DEF SPRITE2,(3,0,1,0,0)=CHR$(213)
85 LOCATE 2,10:PRINT "PUSH ANY KEY TO S
TART !!"
86 IF INKEY$="" THEN 86
87 LOCATE 2,10:PRINT "
         "
90 FOR I=0 TO 3
100 POSITION I,32+I*56,168
110 MOVE I
120 NEXT
130 K1=10
135 LOCATE 0,22:PRINT "HI-SCORE";HI
140 LOCATE 0,23:PRINT"LEVEL";LE;:LOCATE
9,23:PRINT"SCORE";SC;:LOCATE 20,23:PRIN
T "SHIP";SI;
150 TX=RND(4):F1=RND(4):F2=0:MX=0
155 ERA 4:SPRITE 0:SPRITE 1:IF K1<0 TH
EN 400
160 IF F1=3 THEN F2=1
170 SPRITE F2,32+TX*56,32:K1=K1-1:PLAY
"O5C0"
```

```
180 KY$=INKEY$
190 IF KY$="Z" THEN MX=0:GOTO 240
200 IF KY$="X" THEN MX=1:GOTO 240
210 IF KY$="・" THEN MX=2:GOTO 240
220 IF KY$="/" THEN MX=3:GOTO 240
230 T1=T1-1:IF T1<0 THEN 500
235 GOTO 300
240 FOR MY=160 TO 24 STEP -1
250 SPRITE 2,36+MX*56,MY
260 NEXT
270 IF MX=TX THEN 310
300 GOTO 180
310 POSITION 4,32+TX*56,32:SPRITE F2:M
OVE 4:T1=T2
315 IF MOVE(4) THEN 315
320 IF F2=1 THEN SI=SI-1:PLAY"O1C9":IF
SI<0 THEN 600
330 IF F2=0 THEN SC=SC+1*LE
340 GOTO 140
400 LOCATE 6,9:PRINT"CLEAR LEVEL";LE:LO
CATE 5,11:PRINT"TRY NEXT LEVEL !!":PAUS
E 200
410 LOCATE 6,9:PRINT"                  ":LO
CATE 5,11:PRINT"                    "
420 LE=LE+1
430 T2=T2-10:IF T2<30 THEN TI=30
440 GOTO 130
500 T1=T2:IF F2=1 THEN 150
510 FOR I=24 TO 184
520 SPRITE 2,36+TX*56,I
530 NEXT
540 ERA TX:POSITION 4,32+TX*56,168:MOV
E 4:PLAY"O1C9"
```

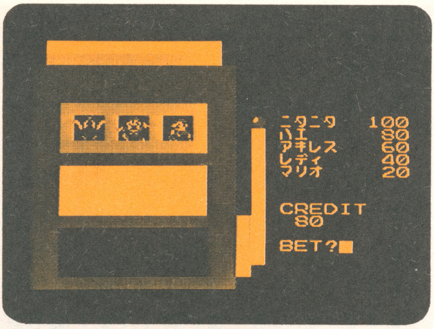
```
545 IF MOVE(4) THEN 545
550 SI=SI-1:IF SI<0 THEN 600
560 MOVE TX
570 GOTO 130
600 LOCATE 9,9:PRINT "GAME OVER"
610 LOCATE 7,11:INPUT "TRY AGAIN ";TR$
620 IF SC>HI THEN HI=SC
630 IF TR$<>"N" THEN 10
640 CLS:SPRITE OFF
650 END
```

●BG-GRAPHICリスト

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0		B10	B50	B50	C10				B10	B50	B50	C10				B10	B50	B50	C10				B10	B50	B50	C10		
1	A10	D40			D40	A00		A10	D40			D40	A00		A10	D40			D40	A00		A10	D40			D40	A00	
2		B40			D00				B40			D00				B40			D00				B40			D00		
3																												
4																												
5																												
6																												
7																												
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17																												
18		G02			G12				G02			G12				G02			G12				G02			G12		
19	I 03	I 13			I 13	I 23		I 03	I 13			I 13	I 23		I 03	I 13			I 13	I 23		I 03	I 13			I 13	I 23	
20	I 43	I 43	I 43	I 43	I 43	I 43		I 43	I 43	I 43	I 43	I 43	I 43		I 43	I 43	I 43	I 43	I 43	I 43		I 43	I 43	I 43	I 43	I 43	I 43	

2 ファミリー・スロットマシン

■ゲームの遊び方



このプログラムはファミリーベーシックに内蔵されているキャラクタを使ったスロットマシンのゲームです。

プログラムを実行させると「BET？」と表示されますので、賭け金をキーボードから入力してください。賭け金は１から10までです。

入力が終り、リターンキーを押すと、賭け金に応じた配当金が表示され、CREDITから賭け金が引かれてスロットマシンがスタートします。[1][2][3]のキーを順に押してスロットを１つずつ止めるか、[3]のキーだけを押して全部のスロットを止めてください。スロットが止まったときに表示されている３つのキャラクタが同じ種類のものであれば、表示されている配当金が得られCREDITに加算されます。CREDITが０になると、「PLAY AGAIN？」と尋ねてきますので、ゲームを行なうときはN以外のキーを押し、リターンキーを押してください。

なお賭け金は10より大きくできませんが、CREDITから賭け金を引いた値が０より小さくなるような賭け方もできませんので注意してください。

■プログラムの説明

10行〜20行	画面、変数の初期設定
30行	初期キャラクタの設定
35〜60行	アニメキャラクタをスプライトに設定
70〜90行	アニメキャラクタの表示
100行	キー入力値の代入
110〜130行	キー入力値の判断
140〜200行	スロットを回転させるための処理
210〜250行	止まったスロットに表示されているアニメキャラクタが同じかどうかの判断

320行	配当金の表示
330行	画面右側の消去
340〜350行	ＣＲＥＤＩＴが残っているかの判断
360〜380行	リプレイ，プログラム終了の処理
800〜840行	配当金の表示
910行	ＣＲＥＤＩＴの表示
920〜950行	賭け金の入力と入力値の処理

●BASICリスト

```
10 CLS:CGSET1,0:SPRITE ON:VIEW:CO=100:
PLAY "TI"
20 F1=0:F2=0:F3=0:RE=1:BE=0:GOSUB 800
30 R1=RND(23)*4:R2=RND(23)*4:R3=RND(23)
*4
35 IF F1=1 THEN 45
40 DEF SPRITE0,(0,1,1,0,0)=CHR$(R1)+CHR$(R
1+1)+CHR$(R1+2)+CHR$(R1+3)
45 IF F2=1 THEN 60
50 DEF SPRITE1,(0,1,1,0,0)=CHR$(R2)+CHR$(R
2+1)+CHR$(R2+2)+CHR$(R2+3)
60 DEF SPRITE2,(0,1,1,0,0)=CHR$(R3)+CHR$(R
3+1)+CHR$(R3+2)+CHR$(R3+3)
70 SPRITE 0,40,80
80 SPRITE 1,64,80
90 SPRITE 2,88,80
100 ST$=INKEY$
110 IF ST$="1" THEN F1=1
120 IF ST$="2" THEN F2=1
130 IF ST$="3" THEN F3=1
140 IF F1=1 THEN 160
150 R1=R1+4:IF R1>92 THEN R1=0
```

```
160 IF F2=1 THEN 180
170 R2=R2+4:IF R2>92 THEN R2=0
180 IF F3=1 THEN 210
190 R3=R3+4:IF R3>92 THEN R3=0
200 PLAY "O5C0":GOTO 35
210 IF (R1=88 OR R1=92) AND (R2=88 OR R2
=92) AND (R3=88 OR R3=92) THEN WI=100
220 IF (R1=56 OR R1=60) AND (R2=56 OR R
2=60) AND (R3=56 OR R3=60) THEN WI=80
230 IF (R1>=64 AND R1<=84) AND (R2>=64
AND R2<=84) AND (R3>=64 AND R3<=84) TH
EN WI=60
240 IF (R1>=28 AND R1<=52) AND (R2>=28
AND R2<=52) AND (R3>=28 AND R3<=52) TH
EN WI=40
250 IF (R1>=0 AND R1<=24) AND (R2>=0 AN
D R2<=24) AND (R3>=0 AND R3<=24) THEN W
I=20
320 LOCATE 17,19:PRINT"YOU GET";RE*WI:C
O=CO+RE*WI:PAUSE 100:WI=0
330 FOR I=0 TO 22:LOCATE 17,I:PRINT
"          ";:NEXT
340 IF CO>0 THEN 20
350 CLS:SPRITE OFF:LOCATE 7,10:PRINT"Y
OU BUST !!"
360 LOCATE 6,12:INPUT"PLAY AGAIN";AG$
370 IF AG$<>"N" THEN RUN
380 CLS:END
800 LOCATE 17,7:PRINT"ニタニタ ";RE*100;
810 LOCATE 17,8:PRINT"ハエ     ";RE*80;
820 LOCATE 17,9:PRINT"アキレス  ";RE*60;
830 LOCATE 17,10:PRINT"レディ    ";RE*40;
840 LOCATE 17,11:PRINT"マリオ     ";RE*20;
```

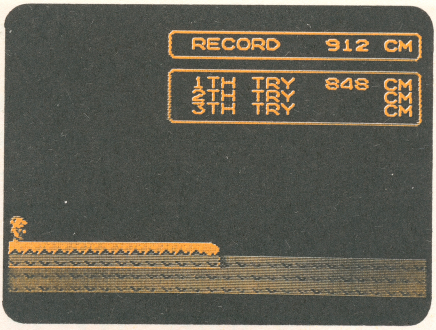
```
910 LOCATE 17,14:PRINT"CREDIT":LOCATE 1
7,15:PRINT CO;"  "
915 IF BE>0 THEN RETURN
920 LOCATE 17,17:INPUT "BET";BE
930 IF BE>10 OR CO-BE<0 THEN 920
940 RE=BE:CO=CO-BE
950 GOTO 800
```

●BG-GRAPHICリスト

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1		M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52															
2		M62	M62	M62	M62	M62	M62	M62	M62	M62	M62	M62	M62															
3	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72														
4	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72														
5	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72														
6	M72	M72	M62	M62	M62	M62	M62	M62	M62	M62	M62	M62	M72	M72														
7	M72	M72	M62			M62			M62			M62	M72	M72		M72												
8	M72	M72	M62			M62			M62			M62	M72	M72		M52												
9	M72	M72	M62	M62	M62	M62	M62	M62	M62	M62	M62	M62	M72	M72		M52												
10	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72		M52												
11	M72	M72	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M72	M72	5	M52												
12	M72	M72	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M72	M72		M52												
13	M72	M72	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M72	M72		M52												
14	M72	M72	M52	M52	M52	M52	M52	M52	M52	M52	M52	M52	M72	M72		M52												
15	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M62	M52												
16	M72	M72											M72	M72	M62	M52												
17	M72	M72											M72	M72	M62	M52												
18	M72	M72											M72	M72	M62	M52												
19	M72	M72											M72	M72	M62													
20	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72	M72														

3 ジャンピング・マリオ

■ゲームの遊び方



崖の上に立っているマリオを崖からジャンプさせ、飛距離を競うゲームです。

プログラムを実行させるとスタートの合図の笛が鳴りますから、コントローラーⅠのA，Bのボタンを押し、マリオをスタートさせてください。A，Bのボタンを押す回数が多ければ多いほどマリオのスピードも速くなってきます。そして、マリオが崖の端までくると自然にジャンプします。

マリオの飛距離は助走のときのスピードが速ければ速いほど伸びます。つまり、A，Bのボタンを押した数が多ければ多いほど遠くへ飛ぶということです。

ジャンプは全部で3回でき、1回ずつ画面に表示されます。3回目のジャンプが終わると、画面に「PUSH START BOTTON！！」と表示されますので、続けてゲームを行ないたいときは、コントローラーⅠのSTARTボタンを押してください。

画面の「RECORD」と書かれているところに今までの最高記録が表示されますので、コントローラーをこわさない程度に記録を競い合ってください。

なお、コントローラーⅠのAまたはBのどちらか1方のボタンを押しっぱなしにしても飛距離はたいして出ないようになっていますのであしからず。

■プログラムの説明

10行	画面の初期設定、配列変数の宣言
20行	変数の初期設定
30〜75行	キャラクタの設定
80〜120行	画面に文字を表示
160行	崖の上にマリオを表示、笛を鳴らす

170行	コントローラー１のＡボタンが押されているかの判断
180～190行	崖の上のマリオを消して動作の初期位置を設定
200行	Ｋにコントローラー１のボタンの状態を代入
205～210行	Ｋの状態による処理
220～230行	動作中のマリオの加速
240行	マリオを動作
250行	マリオが崖の端まで来ているかの判断
260行	ルーチンの終り
300～350行	マリオのジャンプ（上昇）
360～400行	マリオのジャンプ（下降）
410行	スコアを変数に代入
430行	スコアの表示
440行	次のトライのときの変数を決定
450行	３回のトライが終わったかの判断
460行	次のトライへ
500～520行	ハイスコアの判断
530～570行	リプレイの処理

●BASICリスト

```
10 CLS:CGSET1,0:SPRITE ON:VIEW:DIMSC(2)
20 F=0:TI=2000:SP=0
30 DEF MOVE(0)=SPRITE(0,3,4,3,1,0)
40 DEF MOVE(1)=SPRITE(0,2,1,1,1,0)
50 DEF MOVE(2)=SPRITE(0,4,1,1,0,0)
60 DEF MOVE(3)=SPRITE(0,3,3,3,1,0)
65 DEF MOVE(4)=SPRITE(0,3,2,3,1,0)
70 DEF MOVE(5)=SPRITE(0,3,1,3,1,0)
75 DEF SPRITE 0,(0,1,1,1,0)=CHR$(5)+CHR
$(4)+CHR$(7)+CHR$(6)
80 FOR I=0 TO 2
90 LOCATE 11,I+4:PRINT I+1;"TH TRY
```

```
     CM"
100 NEXT
110 LOCATE 12,1:PRINT "RECORD        CM"
120 LOCATE 20,1:PRINT HI
160 SPRITE 0,12,143:PAUSE 100:PLAY "C6:
G6"
170 IF STRIG(0)<>8 THEN 170
180 MOVE 0:SPRITE 0
190 POSITION 0,12,143
200 K=STRIG(0)
205 IF K=4 THEN D=20+A
210 IF K=8 THEN TI=TI-D:D=1
220 IF SP=0 AND TI<1900 THEN POSITION
3,XPOS(0),YPOS(0):SP=3:MOVE SP:ERA 0:A=5
225 IF SP=3 AND TI<1700 THEN POSITION
4,XPOS(3),YPOS(3):SP=4:MOVE SP:ERA 3:A=2
0
230 IF SP=4 AND TI<1400 THEN POSITION
5,XPOS(4),YPOS(4):SP=5:MOVE SP:ERA 4:A=4
0
240 MOVE SP
250 IF XPOS(SP)>112 THEN 300
260 GOTO 200
300 ERA SP
310 POSITION 1,XPOS(SP),YPOS(SP)
320 FOR I=0 TO (20-TI/100)*2
330 MOVE 1
340 IF MOVE(1) THEN 340
350 NEXT
360 ERA 1:POSITION 2,XPOS(1),YPOS(1)
370 FOR I=0 TO (20-TI/100)*2+12
380 MOVE 2
390 IF MOVE(2) THEN 390
400 NEXT
410 PAUSE 150:ERA 2:SC(F)=2000-TI
430 LOCATE 20,F+4:PRINT SC(F)
```

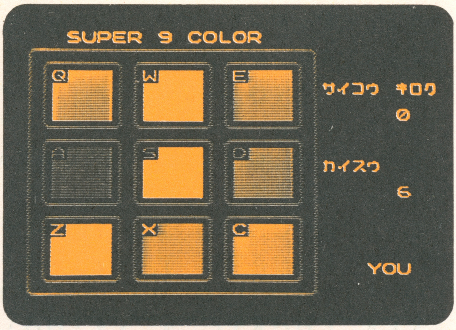
```
440 F=F+1:TI=2000:SP=0:A=0
450 IF F>2 THEN 500
460 GOTO 160
500 FOR I=0 TO 2
510 IF SC(I)>HI THEN HI=SC(I)
520 NEXT
530 LOCATE 20,1:PRINT HI
540 LOCATE 3,12:PRINT"PUSH START BUTTO
N !!":PAUSE 20
550 LOCATE 3,12:PRINT"
    ":PAUSE 20
560 IF STRIG(0)<>1 THEN 540
570 F=0:GOTO 80
```

●BG-GRAPHICリスト

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0											K72	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	L02
1											K62																	K62
2											L12	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	L22
3											K72	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	L02
4											K62																	K62
5											K62																	K62
6											K62																	K62
7											L12	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	K52	L22
8																												
9																												
10																												
11																												
12																												
13																												
14																												
15																												
16																												
17	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I13	I23										
18	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I43	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40
19	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40
20	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40	I40

4 スーパー9(ナイン)カラー

ゲームの遊び方



画面に3×3の合計9個のマス目が描かれていて、それぞれに異なった色がついています。まず画面右下に「COM」と表示され、コンピュータがこの9個の中から任意の1個を選び、音を出して色を点滅させます。次に画面右下に「YOU」と表示されますので、Q W E A S D Z X C の9個のキーでそれに対応しているものを押してください。

押したキーが間違っていなければ、コンピュータがさらに任意の1個を選び、最初に選んだものに続けて音を出して色を点滅させます。さきほどのときと同様にコンピュータのやった通りにキーを押してください。押したキーが間違っていなければ、さらに任意の1個が追加されます。

このようにしてコンピュータは1個ずつ任意の色を選び、最高30個まで増します。ゲームするたびにパターンが変わりますので記憶力だけがたよりです。もし、コンピュータがやった様にキーを押さなかったときは、ブザーが鳴り、ゲームは終ります。画面に「TRY AGAIN?」と表示されますので、もう一度やりたいときは「N」以外のキーを押してからリターンキーを押してください。ハイスコアが記録され、再びゲームが始まります。

このプログラムは残りのメモリが少ないので、内容の変更や追加はしないでください。

プログラムの説明

10行	画面、変数の初期設定
20行	配列変数の宣言
25〜26行	色のデータと音のデータを変数に読み込む
30行	背景との合成、変数の初期設定
35〜50行	パレットの配色番号の色の組み合わせを変える

60〜140行	画面のマス目の色を変更
150〜160行	スコア、ハイスコアの表示
260行	乱数により任意の位置を決定
265行	「ＣＯＭ」の表示
270〜330行	コンピュータ側の音出しと色の点滅
340行	「ＹＯＵ」の表示
350〜460行	キー入力による判断
470行	キー入力による音出しと色の点滅
480行	パターンをクリアしたかの判断
490行	ルーチンの終わり
500行	キー入力が正しいかの判断
510〜560行	音出しと色の点滅のサブルーチン
600〜630行	ゲーム終了の処理
800〜810行	色と音のデータ

●BASICリスト

```
10 CLS:CGSET1,1:SPRITE ON:PLAY"T3":HI=0
20 DIM X(30),Y(30),C(2,2),M$(2,2),F(2,2)
25 FOR I=0 TO 2:FOR J=0 TO 2:READ C(I,
J):F(I,J)=C(I,J):NEXT:NEXT
26 FOR I=0 TO 2:FOR J=0 TO 2:READ M$(I,
J):NEXT:NEXT
30 VIEW:SC=0:T=100:E=-1:D=0
35 FOR I=0 TO 2
40 PALETB I+1,15,C(I,0),C(I,1),C(I,2)
50 NEXT
60 FOR I=0 TO 2
80 FOR J=0 TO 2
90 COLOR 2+6*J,3+6*I,I+1
100 COLOR 4+6*J,3+6*I,I+1
110 COLOR 2+6*J,6+6*I,I+1
```

```
120 COLOR 4+6*J, 6+6*I, I+1
130 NEXT
140 NEXT
150 LOCATE 3, 0:PRINT "SUPER 9 COLOR"
160 LOCATE 20,4:PRINT"サイコウ キロク":LOCATE 2
4, 6:PRINT HI:LOCATE 20, 10:PRINT"カイスウ":L
OCATE 24, 12:PRINT SC
260 X(D)=RND(3):Y(D)=RND(3)
265 LOCATE 23, 18:PRINT "COM":PAUSE 100
270 FOR I=0 TO D
280 Z=X(I):W=Y(I):GOSUB 510
330 NEXT
340 D=D+1:LOCATE 23, 18:PRINT "YOU"
350 K$=INKEY$
360 IF K$="Q" THEN A=0:B=0
370 IF K$="W" THEN A=0:B=1
380 IF K$="E" THEN A=0:B=2
390 IF K$="A" THEN A=1:B=0
400 IF K$="S" THEN A=1:B=1
410 IF K$="D" THEN A=1:B=2
420 IF K$="Z" THEN A=2:B=0
430 IF K$="X" THEN A=2:B=1
450 IF K$="C" THEN A=2:B=2
460 IF K$="" THEN 350
470 E=E+1:Z=A:W=B:GOSUB 500
480 IF D=E+1 THEN E=-1:SC=SC+1:GOTO 16
0
490 GOTO 350
500 IF A<>X(E) OR B<>Y(E) THEN 600
510 F(Z, W)=13
520 PALETB Z+1, 15, F(Z, 0), F(Z, 1), F(Z, 2)
530 PLAY M$(Z, W)
540 F(Z, W)=C(Z, W)
```

```
550 PALETB Z+1, 15, F(Z, 0), F(Z, 1), F(Z, 2)
560 RETURN
600 PLAY"O1C9":CLS:LOCATE 8, 10:INPUT"TR
Y AGAIN ";TR$
610 IF SC>HI THEN HI=SC
620 IF TR$<>"N" THEN 30
630 CLS:CGSET 1, 1:END
800 DATA 41, 39, 28, 26, 48, 22, 37, 17, 20
810 DATA O2C3, O2D3, O2E3, O2F3, O2G3, O2A3,
O2B3, O3C3, O3D3
```

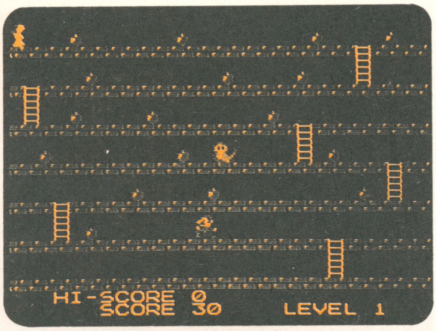
●BG-GRAPHICリスト

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1	K70	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	L00								
2	K60	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K60								
3	K60	K60	Q	M52	M52	M52	K60	K60	W	M62	M62	M62	K60	K60	E	M72	M72	M72	K60	K60								
4	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
5	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
6	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
7	K60	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	K60								
8	K60	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K60								
9	K60	K60	A	M52	M52	M52	K60	K60	S	M62	M62	M62	K60	K60	D	M72	M72	M72	K60	K60								
10	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
11	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
12	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
13	K60	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	K60								
14	K60	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K70	K50	K50	K50	K50	L00	K60								
15	K60	K60	Z	M52	M52	M52	K60	K60	X	M62	M62	M62	K60	K60	C	M72	M72	M72	K60	K60								
16	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
17	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
18	K60	K60	M52	M52	M52	M52	K60	K60	M62	M62	M62	M62	K60	K60	M72	M72	M72	M72	K60	K60								
19	K60	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	L10	K50	K50	K50	K50	L20	K60								
20	L10	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	K50	L20								

5 森田さんちのマリオくん

注：このプログラムはV.2.0以降で使えます

■ゲームの遊び方



マリオがビルの１階にいますのでコントローラーを使って、落ちているリンゴを拾いながら７階のレディのところまで持っていってください。

リンゴは１個10点でいくつ拾ってもかまいませんが、１階にいるファイヤーボールが７階のレディのところに行くまでに持っていかないとゲームオーバーになってしまいます。

画面の両端はワープトンネルになっていて、マリオが反対側に出られますが、上の階に行くにはハシゴを使わなければなりません。

リンゴやハシゴの位置を変えて自分の面を作ることもできます。

■プログラムの説明

10行	画面消去、画面初期設定、ハイスコアのクリア
12〜15行	変数初期設定、画面の合成
20行	マリオのキャラクタを設定
22行	ハイスコアの表示
25行	スコア、レベルの表示
100行	マリオを表示
112行	マリオが画面の左端へ来たら400行へ
115行	ゲームオーバーの判断
120行	コントローラーⅠが左側に押されたときの処理
130行	コントローラーⅠが右側に押されたときの処理
133行	マリオの位置をＢＧ　ＧＲＡＰＨＩＣ画面位置に変換
134行	マリオの位置にハシゴがあるのかの判断
135行	コントローラーⅠが上側に押されたときの処理
140行	マリオの位置をＢＧ　ＧＲＡＰＨＩＣ画面位置に変換

150行	マリオの位置にリンゴがあるかの判断
160行	マリオの位置にハシゴがあるかの判断
170行	ルーチンの終わり
180行	マリオを上側に動かす
190行	動作が完了していなかったら190行へ
200行	ルーチンの終わり
210行	スコアの加算、表示
300行	ハイスコアの判断、表示
305～310行	ゲームオーバーの表示
320行	リプレイの表示と処理
330行	プログラム終了の処理
400～420行	レベルクリア時の処理

●BASICリスト

```
10 CLS:CGSET1,0:SPRITEON:HS=0
12 L=5:LE=1:SC=0
15 MX=10:MY=168:M=1:FY=192:VIEW:PLAY "O
5T1:T1:T1"
20 DEFMOVE(0)=SPRITE(0,7,1,3,0,0):DEFMOV
E(1)=SPRITE(0,3,1,3,0,0):DEFMOVE(2)=SPRI
TE(0,1,1,12,0,0)
22 LOCATE 3,21:PRINT "HI-SCORE";HS;
25 LOCATE 6,22:PRINT "SCORE";SC;:LOCATE
18,22:PRINT "LEVEL";LE;
30 DEFMOVE(3)=SPRITE(5,7,L,120,0,3):DEFS
PRITE0,(2,1,0,1,0)=CHR$(33)+CHR$(32)+CHR
$(35)+CHR$(34)
100 POSITIONM,MX,MY:MOVEM:SPRITE0,13,24
110 IF MOVE(3)=0 THEN ERA3:FY=FY-24:POS
ITION3,240,FY:MOVE3
112 IF XPOS(M)<28 AND YPOS(M)=24 THEN 400
```

```
115 IF XPOS(3)<24 AND YPOS(3)=24 THEN CU
T 3:SPRITE0:GOTO 300
120 IF STICK(0)=1 THEN ERA0,2:POSITION
1,XPOS(M), YPOS(M):M=1:MOVEM
130 IF STICK(0)=2 THEN ERA1,2:POSITION
0,XPOS(M),YPOS(M):M=0:MOVEM
133 MX=(XPOS(M)-16)/8+1:MY=(YPOS(M)-24)/
8+1:IF MX>27 OR MX<0 THEN 170
134 IF SCR$(MX,MY)=CHR$(&HD7) THEN 210
135 IF STICK(0)=8 AND SCR$(MX,MY)=CHR$(&H
D2) THEN 160
140 MX=(XPOS(M)-16)/8+1:MY=(YPOS(M)-24)/
8+1:IF MX>27 OR MX<0 THEN 200
150 IF SCR$(MX,MY)=CHR$(&HD7) THEN 210
160 IF STICK(0)=8 AND SCR$(MX,MY)=CHR$(&H
D2) THEN 180
170 GOTO 110
180 POSITION2,XPOS(M),YPOS(M):ERA0,1:M=2:
MOVE2
190 IF MOVE(2) THEN 190
200 GOTO 170
210 LOCATE MX,MY:PRINT" ";:SC=SC+10:PLA
Y"C0":LOCATE6,22:PRINT"SCORE";SC:GOTO 17
0300 IF SC>=HS THEN HS=SC:LOCATE 3,21:
PRINT "HI-SCORE";HS;
305 FOR I=0 TO 5:LOCATE 6,9+I:PRINT "
          ":NEXT
310 LOCATE 8,10:PRINT "GAME OVER":LOCAT
E 8,11
320 LOCATE 7,12:INPUT"TRY AGAIN ";RP$:I
F RP$<>"N" THEN 12
```

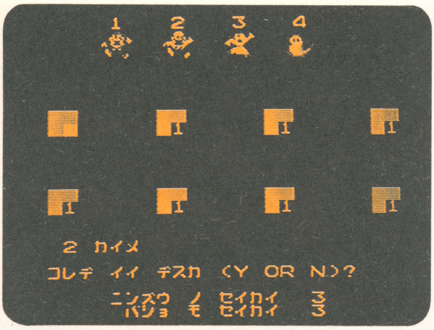
```
330 CGSET1,1:CLS:END
400 FOR I=0 TO 4:LOCATE 4,9+I:PRINT"
        ":NEXT
405 CUT0,1,2,3:LOCATE 6,10:PRINT"LEVEL";
LE;" CLEAR !!";:PAUSE 200:LE=LE+1:L=L-
1:IF L<1 THEN L=1
410 LOCATE 5,12:PRINT"CHALENGE LEVEL";L
E;"!!":PAUSE 200
420 ERA 0,1,2,3:GOTO 15
```

●BG-GRAPHICリスト

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0																												
1					H72							H72									H72					H72		
2	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	H22	F32	F32	F32	F32
3																								H22				
4						H72									H72					H72				H22				
5	F32	H22	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
6		H22																										
7		H22			H72					H72								H72										
8	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	H22	F32	F32	F32	F32	F32	F32	F32	F32
9																				H22								
10			H72									H72								H22		H72						
11	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	H22	F32	F32
12																										H22		
13									H72					H72										H72		H22		
14	F32	F32	F32	H22	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32
15				H22																								
16				H22		H72								H72														
17	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	H22	F32	F32	F32	F32	F32	F32
18																						H22						
19					H72															H72		H22						
20	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32	F32

⑯ マリオを探せ!

ゲームの説明



画面にある8つのドアのむこう側にはマリオ、ルイージ、レディ、ファイアーボールが隠れています。どのドアに誰が隠れているのか当ててください。

ただし、マリオ、ルイージ、レディ、ファイアーボールはそれぞれ何人ずついるのかわかっていません。考えながらやらないとなかなか全部を当てることはできません。

プログラムを実行させると、まず8つのドアが出てきて、ドアの右下でカーソルが点滅します。あなたが予想した番号を画面の上にでているサンプルを参考に入力してください。1つのドアに入力が終わると次のドアにカーソルが移ります。全部のドアに入力が終わるとこれでいいのか聞いてきますから、修正したい場合は「N」、そのままでいいときは他のキーを押します。

すると、ドアのむこうに隠れているそれぞれのキャラクタの数と入力したそれぞれのキャラクタの番号が場所に関係なく合計でいくつ正解しているのかと、キャラクタの番号と場所も正解しているのが合計でいくつなのかを数えてくれます。そしてカーソルは再びドアのところに移りますので、これをヒントに当たるまで考えてください。

なお、カーソルが点滅しているときに1～4以外のキーを押すとすべて0になりますし、CLR や → なども本来の機能が使えずに、0として入力されます。これは画面がくずれないようにするために行っています。

プログラムの説明

10行	画面、変数の初期設定
20行	配列変数の宣言
30～110行	ドアに隠れているキャラクタの設定

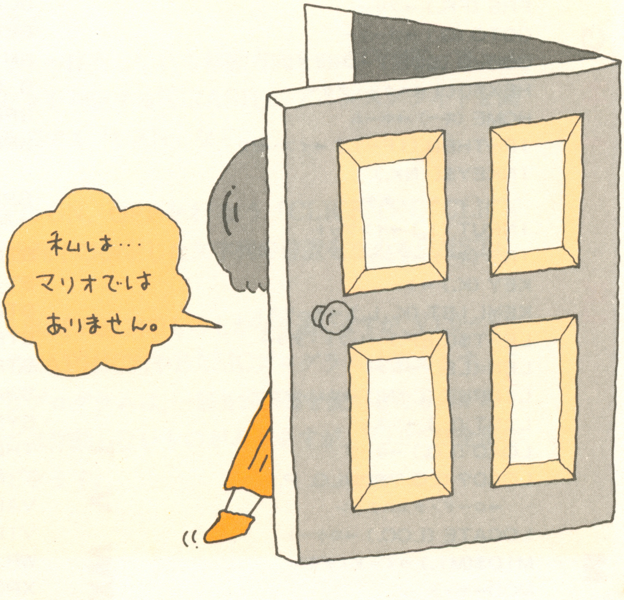
120～150行	サンプルキャラクタの設定
160～190行	サンプルキャラクタの表示
200～220行	ドアの表示
230行	回数の表示
234～250行	カーソルの表示とキー入力
260～270行	修正するかの判断と処理
300～346行	正解数を求める
350～360行	正解数の表示
370行	全部正解しているかの判断
380～390行	全部正解でなかったときの処理
550～590行	ドアの後ろにキャラクタを表示
600～620行	ドアを消す
630～640行	プログラム終了の処理

●BASICリスト

```
10 CLS:CGSET 1,0:SPRITE ON:VIEW:F=1
20 DIM C(7),P(7),K$(7),K(7)
30 FOR I=0 TO 7
40 C(I)=RND(4)
50 IF C(I)=0 THEN P(I)=0:M1=M1+1
60 IF C(I)=1 THEN P(I)=0:M2=M2+1
70 IF C(I)=2 THEN P(I)=1:M3=M3+1
80 IF C(I)=3 THEN P(I)=5:M4=M4+1
100 DEF MOVE(I)=SPRITE(P(I),0,1,1,1,C(I))
110 NEXT
120 DEF SPRITE 0,(0,1,0,0,0)=CHR$(24)+CHR
$(25)+CHR$(26)+CHR$(27)
130 DEF SPRITE 1,(1,1,0,0,0)=CHR$(24)+CHR
$(25)+CHR$(26)+CHR$(27)
140 DEF SPRITE 2,(2,1,0,0,0)=CHR$(52)+CHR
$(53)+CHR$(54)+CHR$(55)
150 DEF SPRITE 3,(3,1,0,0,0)=CHR$(116)+CH
R$(117)+CHR$(118)+CHR$(119)
160 FOR I=0 TO 4
```

```
170 SPRITE I,I*32+68,32
180 NEXT
190 LOCATE 7,0:PRINT"1      2      3      4"
200 FOR I=0 TO 7
205 IF I>3 THEN H=I-4:G=6:GOTO 210
206 H=I
210 LOCATE H*7+3,7+G:PRINT CHR$(254);CHR$(254):LOCATE H*7+3,8+G:PRINT CHR$(254);CHR$(254)
220 NEXT
230 LOCATE 3,17:PRINT F;"カイメ";" "
234 FOR I=0 TO 7
235 IF I>3 THEN W=I-4:Z=6:GOTO 240
236 W=I
240 LOCATE W*7+4,8+Z:K$(I)=INKEY$(0):LOCATE W*7+4,8+Z:PRINT K$(I)
250 NEXT
260 LOCATE 3,19:PRINT "コレデ イイ デスカ (Y OR N)?";:S$=INKEY$(0)
270 IF S$="N" THEN Z=0:GOTO 230
280 FOR I=0 TO 7
290 K(I)=VAL(K$(I)):IF K(I)=C(I)+1 THEN A1=A1+1
320 IF K(I)=1 AND B1<M1 THEN B1=B1+1
330 IF K(I)=2 AND B2<M2 THEN B2=B2+1
340 IF K(I)=3 AND B3<M3 THEN B3=B3+1
345 IF K(I)=4 AND B4<M4 THEN B4=B4+1
346 NEXT
350 LOCATE 7,21:PRINT"ニンズウ ノ セイカイ ";B1+B2+B3+B4
360 LOCATE 8,22:PRINT"バショ モ セイカイ ";A1
370 IF A1=8 THEN 500
380 B1=0:B2=0:B3=0:B4=0:F=F+1:Z=0:A1=0
390 GOTO 230
550 FOR I=0 TO 7
```

```
560 IF I>3 THEN X=I-4:Y=48:GOTO 580
570 X=I
580 POSITION I,X*56+40,79+Y:MOVE I
590 NEXT
600 FOR I=7 TO 14
610 LOCATE 3,I:PRINT"
       "
620 NEXT
630 LOCATE 10,11:PRINT"セイカイ !!":PAUSE 3
00
640 CLS:SPRITE OFF:END
```



ベーシックの命令やメッセージの読み方

A

ABS〔AB.〕→アブス
ASC〔AS.〕→アスキー

B

BEEP〔B.〕→ビープ

C

CALL〔CA.〕→コール
CGEN〔CGE.〕→シージェン
CGSET〔CG.〕→シー・ジー・セット
CHAR→キャラ
CHR$〔CH.〕→キャラクター・ダラー
CLEAR〔CLE.〕→クリア
CLS〔CL.〕.→クリア・スクリーン
COLOR〔COL.〕→カラー
CONT〔C.〕→コント
COPY→コピー
CSRLIN〔CS.〕→カーソル・ライン
CTR（キー）→コントロール
CUT〔CU.〕→カット

D

DATA〔D.〕→データ
DEF MOVE〔DE. M.〕
→デフ・ムーブ
DEF SPRITE〔DE. SP.〕
→デフ・スプライト
DEL（キー）→デリート
DIM〔DI.〕→ディム

E

END〔E.〕→エンド
ERA〔ER.〕→エラー
ESC（キー）→エスケープ

F

FILE→ファイル
FOR〜TO〜STEP〔F.−TO−ST.〕
→フォ〜ツー〜ステップ
FRE〔FR.〕→フリー

G

GOTO〔G.〕→ゴーツー
GOSUB〔GOS.〕→ゴーサブ

H

HEX$〔H.〕→ヘクス・ダラー
HOME（キー）→ホーム

I

IF〜THEN〔IF−T.〕→イフ〜ゼン
INKEY$〔INK.〕
→イン・キー・ダラー
INPUT〔I.〕→インプット
INS（キー）→インサート

K L

KEY〔K.〕→キー
KEYLIST〔K. L.〕→キーリスト
LEFT$〔LEF.〕→レフト・ダラー
LEN〔LE.〕→レン
LINPUT〔LIN.〕→エル・インプット
LIST〔L.〕→リスト
LOAD〔LO.〕→ロード
LOAD?〔LO.?またはLO. P.〕
→ロード・クエスチョン
LOCATE〔LOC.〕→ロケート

M

MID$〔MI.〕→ミッド・ダラー
MOD→モード
MODE→モード
MOVE〔M.〕→ムーブ
MOVE（n）〔M.（n）〕→ムーブ

N

NEW→ニュー
NEXT〔N.〕→ネクスト

O

O・M ERROR
→オーバー・メモリー・エラー
ON〜〔O.〕→オン〜
O・V ERROR
→オーバー・フロー・エラー
OPALET〔PAL. B PAL. S〕
→パレット

P

PAUSE〔PA.〕→ポーズ
PEEK〔PE.〕→ピーク
PLAY〔PL.〕→プレイ
POKE〔PO.〕→ポーク
POS→ポス
POSITION〔POS.〕→ポジション
PRINT〔?またはP.〕→プリント

R

READ〔REA.〕→リード
READ端子→リードたんし
REM〔 '（アポストロフィ）〕→レム
RESTORE〔RES.〕→リストア
RETURN〔RE.〕→リターン
RIGHT$〔RI.〕→ライト・ダラー
RND〔RN.〕→ランド
RUN〔R.〕→ラン

S

SAVE〔SA.〕→セーブ
SCR$〔SC.〕
→エス・シー・アール・ダラー
SGN〔SG.〕→エス・ジー・エヌ
SELECT→セレクト
SHIFT→シフト（キー）
S・N ERROR→シンタックス・エラー
SPRITE〔SP.〕→スプライト
SPRITE OFF〔SP. OF.〕
→スプライト・オフ
SPRITE ON〔SP. O.〕
→スプライト・オン
STEP〔ST.〕→ステップ
STICK〔STI.〕→スティック
STOP〔STO.〕→ストップ
STRIG〔STRI.〕→エス・トリグ
STR$〔STR.〕→ストリング・ダラー
SWAP〔SW.〕→スワップ
SYSTEM〔S.〕→システム

T

THEN〔T.〕→ゼン
T・P ERROR→テープ・リード・エラー

V

VAL〔VA.〕→バル
VIEW〔V.〕→ビュー

W X Y

WRITE端子→ライトたんし
XPOS〔XP.〕→エックス・ポス
YPOS〔YP.〕→ワイ・ポス

バージョン3.0の特徴

ファミリーベーシックには、いくつかのバージョン（型式）があります。その中のバージョン3.0(V3.0)について、簡単に説明しましょう。新しいだけに、より便利なベーシックパソコンになりました。

①メモリ容量が2倍になった。

②BASICの命令が追加された。

③エディット型ゲームプログラムが内蔵された。

などです。V3.0以前のファミリーベーシックをお持ちの人のために②について機能別にV3.0の内容を挙げてみましょう。

Ⅰ）プログラムの入力・編集

行番号が自動的に付く「AUTO」、全行番号を一度に変更できる「RENUM」、複数の行を削除できる「DELETE」、変更したい文字列を含む行を自動的に探す「FIND」など。

Ⅱ）プログラムの実行

実行中のプログラムのトレースが行なえる「TRON、TROFF」、エラー発生による分岐「ON　ERROR　GOTO、RESUME、ERR、ERL」など。

Ⅲ）BG GRAPHIC関係

2面持つBG GRAPHIC面を切り換える「SCREEN」、BASICからBG GRAPHIC面をセーブ、ロードする「SAVES、LOADS」、BG GRAPHIC面をバッテリ・バックアップするための「BGGET、BGPUT」など。

Ⅳ）スプライト関係

アニメスプライトが保有している移動方向を知る「VCT」、アニメスプライトの重なりを判定する「CRASH」、アニメスプライトを同時に複数個、未定義する「CAN」など。

Ⅴ）その他

キーの入力音のON、OFFを行なう「CLICK ON、CLICK OFF」、任意の文字列中の指定した文字列の位置を求める「INSTR」など。

このように、V3.0はそれ以前のものとかなり違いますが、V1.0～V2.1Aで作ったBG GRAPHIC面やBASICプログラムは使えます。この本に載っているプログラムは、一部分SCR$を使用したものがありますので、V1.0では作動しないものがありますが、V3.0では全部作動させることができます。

著者略歴
森田　将史（もりた　まさし）
昭和60年 3月武蔵大学経済学部卒。
㈱中部相互銀行勤務。十代よりパソコン専門誌に執筆。
パソコンのハード、ソフトに関する記事等多数。
プログラミングとバイクが趣味。
現住所　静岡県静岡市竜南１－437－４（〒420）

ファミリーコンピュータ™
ファミリーベーシック™
PERFECT BOOK

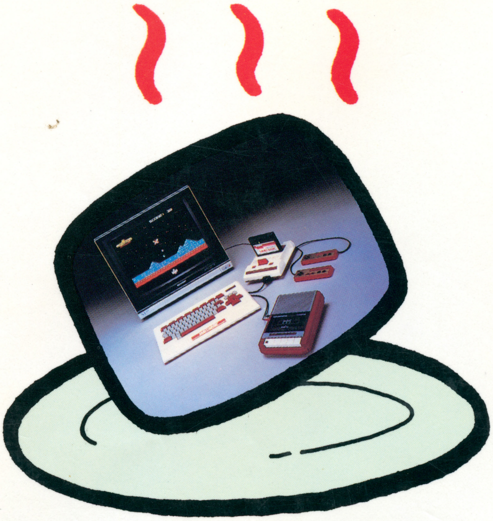
著　者　森田将史

発行者　田村正隆

発行所　株式会社　ナツメ社
東京都千代田区神田神保町１－52（〒101）
電話　03（291）1257
振替　東京３－58661
印　刷　ラン印刷社
製　本　三修紙工

Printed in Japan

本書の内容に関するお問い合わせは、以下の電話番号でお受けいたします。
株式会社ナツメ社　郵便番号101
東京都千代田区神田神保町1-52
電話〈03〉291-1257



ナツメ社　　定価＝980円

ISBN4-8163-0463-0 C2054 ¥980E

ファミリーコンピュータ™&ベーシック™パーフェクトBOOK

ナツメ社

ファミリー コンピュータ™
ファミリー ベーシック™
PERFECT BOOK

発行――1985年4月15日
著者――森田将史
発行者――田村正隆
発行所――株式会社ナツメ社
郵便番号101
東京都千代田区神田神保町1－52
電話〈03〉291－1257
振替　東京3－58661
〈落丁・乱丁本はお取り替えします〉
定価――980円

本書を読まれる方へ

ファミリーベーシックは、いま人気絶好調のゲームマシン、任天堂のファミリーコンピュータに接続して使えるパソコンです。

「おもちゃみたいだ」「パソコンのような気がしない」という声もあります、でも、実際にさわってみるとわかるのですが、ファミリーベーシックは正真正銘、ベーシックが走る立派なパソコンなのです。

ファミリーベーシックの特長といえば、その機能がゲームを簡単に作ることができるように設計されていることです。ゲーム作りのときの背景やキャラクタ作りにかかるわずらわしさが少なくなっています。ですから、初心者の方でもゲームをプログラムできるのです。

本書は、このようなファミリーベーシックの機能を生かして、楽しくカンタンにゲーム作りができるようになっています。

初めてベーシックを学ばれる方は、最初にとまどいがあるはずですが、少し読み進めば、すぐにプログラムが理解できるようになるはずです。

また、この本を開けばファミリーコンピュータのゲームの世界が見えてきます。巻末部分には、打ち込めばすぐに遊べるプログラムが6本用意されていますし、これも、お楽しみの一つです。

COVER DESIGN/ILLUSTRATION YOSHIYUKI co., ltd.

任天堂

FAMILY COMPUTER™
ファミリーコンピュータ™
FAMILY BASIC™
AND ファミリーベーシック™

PERFECT BOOK

ファミリーコンピュータ™&ベーシック™ パーフェクトBOOK

ナツメ社